Vojnotehnicki glasnik/Military Technical Courier
ISSN: 0042-8469
vojnotehnicki.glasnik@mod.gov.rs
University of Defence
Serbia

Proti, Danijela D.
INFLUENCE OF PRE-PROCESSING ON ANOMALY-BASED INTRUSION DETECTION
Vojnotehnicki glasnik/Military Technical Courier, vol. 68, núm. 3, julio-septiembre, 2020, pp. 598-611
University of Defence

INFLUENCE OF PRE-PROCESSING ON ANOMALY-BASED INTRUSION DETECTION

Danijela D. Protić
Serbian Armed Forces, General Staff,
Department for Telecommunication and Informatics (J-6),
Center for Applied Mathematics and Electronics,
Belgrade, Republic of Serbia,
e-mail: adanijela@ptt.rs,
ORCID iD: https://orcid.org/0000-0003-0827-2863

DOI: 10.5937/vojtehg68-27319; https://doi.org/10.5937/vojtehg68-27319

FIELD: Computer Sciences
ARTICLE TYPE: Original scientific paper

Abstract:

Introduction/purpose: The anomaly-based intrusion detection system detects intrusions based on a reference model which identifies the normal behavior of a computer network and flags an anomaly. Machine-learning models classify intrusions or misuse as either normal or anomaly. In complex computer networks, the number of training records is large, which makes the evaluation of the classifiers computationally expensive.

Methods: A feature selection algorithm that reduces the dataset size is presented in this paper.

Results: The experiments are conducted on the Kyoto 2006+ dataset and four classifier models: feedforward neural network, k-nearest neighbor, weighted k-nearest neighbor, and medium decision tree. The results show high accuracy of the models, as well as low false positive and false negative rates.

Conclusion: The three-step pre-processing algorithm for feature selection and instance normalization resulted in improving performances of four binary classifiers and in decreasing processing time.

Key words: anomaly-based intrusion detection, machine learning, Kyoto 2006+.

Introduction

Intrusion detection systems (IDSs) monitor computer network behavior to perform diagnostics of the security status and protect the network from malicious activities or various anomalies. Intrusion detection systems can be divided into two basic groups. Misuse or signature based IDSs detect malware based on knowledge accumulated from known attacks. Anomaly based IDSs detect deviations from a model of usual network behavior. The goal of anomaly detection is to build a statistical model of normal network behavior and look for activities which deviate from the created model (Protić & Stanković, 2020, p.7). The main disadvantage of the signature based IDS is a difficulty to detect unknown attacks. The biggest challenge in anomaly detection is to identify what is considered normal. Machine learning (ML) based binary classifiers can detect anomalies with a high accuracy of prediction. In supervised learning, the number of training instances collected over a period of time can be large, which makes the evaluation of the models computationally expensive. Feature selection reduces the training set, which speeds up the processing time and increases the accuracy of the classifiers. This paper shows the results of the experiments of the three-step feature selection and instances normalization pre-processing algorithm conducted on the Kyoto 2006+ dataset and four machine learning models, namely: feedforward neural network (FNN), k – nearest neighbor (k-NN), weighted k-NN (wk-NN), and medium decision tree(DT). Accuracy (ACC), false positive rate (FPR), false negative rate (FNR), and processing time are given.

Feature selection and instances normalization: Three-step pre-processing algorithm

One of the major issues in supervised ML is a large number of instances in the training set. The aim of feature selection is to reduce the dataset size and remove irrelevant features. Furthermore, raw data have to be pre-processed before being fed to the input of the model so that effects of one feature cannot dominate the others. In this paper, a three-step pre-processing algorithm for feature selection is presented. The algorithm is given as follows:

1 Identify and discard all irrelevant features;
2 Remove features which cannot be normalized into the range [-1,1];
3 Normalize instances into the range [-1,1] by applying the hyperbolic tangent function:

$$\tanh(n) = \frac{2}{1+e^{2n}} - 1, \qquad (1)$$

where n is the number of instances in the dataset.

Feature selection improves performances of the classifier, saves memory space and decreases processing time. Additionally, instances normalization speeds up the model training and reduces the domination of one feature over the other ones.

Data collection: The Kyoto 2006+ dataset

The Kyoto 2006+ dataset contains records of real network traffic data collected from November 2006 to December 2015 at five different computer networks inside and outside the Kyoto University (Takakura, 2020) (Protić, 2018, pp.587-589). During the observation period, over 50 million sessions of normal traffic, 43 million sessions of known attacks and 426 thousand sessions of unknown attacks were recorded (Protić & Stanković, 2018, p.44). The dataset consists of 14 statistical features derived from the KDD Cup '99 dataset (Table 1) and 10 additional features which enable more efficient investigation (Table 2) (Ashok Kumar & Venugopalan, 2018), (Song et al, 2011, pp.29-36).

Table 1 – The Kyoto 2006+ Dataset – first 14 features
Таблица 1 – Набор данных Kyoto 2006+ - 14 первых функций
Табела 1 – Kyoto 2006+ база података – првих 14 атрибута

No	Feature	Description
1	Duration	The length of the connection (seconds).
2	Service	The connection's server type (dns, ssh, other).
3	Source bytes	The number of data bytes sent by the source IP address.
4	Destination bytes	The number of data bytes sent by the destination IP address.
5	Count	The number of connections whose source IP address and destination IP address are the same to those of the current connection in the past two seconds.
6	Same_srv_rate	% of connections to the same service in the Count feature.
7	Serror_rate	% of connections that have 'SYN' errors in the Count feature.
8	Srv_serror_rate	% of connections that have 'SYN' errors in the Srv_count (% of connections whose service type is the same to that of the current connections in the past two seconds) feature.
9	Dst_host_count	Among the past 100 connections whose destination IP address is the same to that of the current connection, the number of connections whose source IP address is also the same to that of the current connection.
10	Dst_host_srv_count	Among the past 100 connections whose destination IP address is the same to that of the current connection, the number of connections whose service type is also the same to that of the current connection.
11	Dst_host_same_src_port_rate	% of connections whose source port is the same to that of the current connection in the Dst_host_count feature.
12	Dst_host_serror_rate	% of connections that have 'SYN' errors in the Dst_host_count feature.
13	Dst_host_srv_serror_rate	% of connections that have 'SYN' errors in the Dst_host_srv_count feature.
14	Flag	The state of the connection at the time the connection was written (tcp, udp).

Table 2 – The Kyoto 2006+ Dataset – additional 10 features
Таблица 2 – Набор данных Kyoto 2006+ - 10 дополнительных функций
Табела 2 – Kyoto 2006+ база података – додатних 10 атрибута

No	Feature	Description
1	IDS_detection	Reflects if IDS triggered an alert for the connection; ‘0’ means any alerts were not triggered and an arabic numeral means the different kind of alerts. The arenthesis indicates the number of the same alert.
2	Malware_detection	Indicates if malware, also known as malicious software, was observed at the connection; ‘0’ means no malware was observed, and string indicates the corresponding malware was observed at the connection. The parenthesis indicates the number of the same malware.
3	Ashula_detection	Means if shellcodes and exploit codes were used in the connection; ‘0’ means neither shellcode nor exploit code were observed, and an arabic numeral means the different kinds of the shellcodes or exploit codes. The parenthesis indicates the number of the same shellcode or exploit code.
4	Label	Indicates whether the session was attack or not; ‘1’ means normal. ‘-1’ means a known attack was observed in the session, and ‘-2’ means an unknown attack was observed in the session.
5	Source_IP_Address	Means the source IP address used in the session. The original IP address on IPv4 was sanitized to one of the Unique Local IPv6 Unicast Addresses. Also, the same private IP addresses are only valid in the same month; if two private IP addresses are the same within the same month, it means their IP addresses on IPv4 were also the same, otherwise are different.
6	Source Port Number	Indicates the source port number used in the session.
7	Destination IP Address	It was also sanitized.
8	Destination Port Number	Indicates the destination port number used in the session.
9	Start Time	Indicates when the session was started.
10	Duration	Indicates how long the session was being established.

The proposed algorithm discards all categorical features as well as features for further investigation, excluding the Label feature (step 1), cuts all features containing instances that cannot be normalized into the range [-1,1] (step 2) and normalize the rest of instances (step 3). Out of 24 features of the Kyoto 2006+ data set, 17 features are left after the first pre-processing algorithm step and nine features (5-13) are left after the pre-processing is done. The Label feature is used for the detection of anomalies. Scaled instances not only reduce the effects of one feature to the others but speed up the FNN since the network training is more efficient if normalization is performed on inputs. If the number of inputs is ≥3, the sigmoid functions used in the hidden layer become easily saturated. If the saturation happens at the beginning of the training, the gradients will be small which may slow down the network training (Protić & Stanković, 2020, p.9).Also, instances are scaled due to the fact that distances in the wk-NN model lose accuracy because of a small difference between the farthest and the nearest neighbors.

Classifier Models

In supervised machine learning, classifiers can be divided into two groups. Lazy learners, such as the k-NN and the wk-NN, do not focus on constructing a general model, but store the training data and weight until a test set appears. Eager learners, such as the FNN, construct a classification model before getting data for predictions.

k-Nearest Neighbor

The k-NN stores all instances corresponding to the training data into the *n*-dimensional space. Classification is computed on a simple majority vote of the k-NN of each point, based on the Euclidean distance measure given with (Protić & Stanković, 2020, p.9):

$$d(x_i, y_i) = \sqrt{\sum_{s=1}^{p} (x_{is} - y_{is})^2} \ . \qquad (2)$$

The prediction speed of the k-NN is medium as well as memory usage. Interpretability of the classifier is hard. In the experiments, the distinction between classes is medium, and the number of neighbors is set to 10 (Protić & Stanković, 2018, p.48).

Weighted k-Nearest Neighbor

The main idea of the wk-NN is to extend the k-NN so that the instances within the training set which are particularly close to the new instance should get a higher weight in the decision than more distant ones (Tsigkritis et al, 2018, pp.70-84). The distances are transformed into the weights as follows:

$$w = \frac{1}{d(\mathbf{x}, \mathbf{y})^2} \ . \qquad (3)$$

The wk-NN classifier adapts as the new training data is collected, which allows the algorithm to respond quickly to changes in the input during real-time use. In contrast with the fast training stage, the algorithm requires expensive testing. All the cost of the algorithm is in the processing time. All characteristics of the classifier are the same as for the k-NN model except flexibility which is also medium (medium distinctions between classes using a distance weight) (Protić & Stanković, 2018, p.48).

Medium Decision Tree

The Decision Tree is one of the graph-like algorithms which use branching methods to illustrate every possible outcome of decisions, where nodes represent features, links represent decision rules and leafs represent outcomes. The Iterative Dichotomy 3 algorithm (ID3) calculates entropy and information gain to build a tree (Protić & Stanković, 2020, p.9). Entropy is a measure which controls how the tree decides to split the data. If the target feature can take on *k* different values, then the entropy of the *S* relative to this *k*-wise classification is given as follows:

$$Entropy\ (S) = -\sum_{i=1}^{k} p_i \log_2 (p_i), \tag{4}$$

where p_i represents the proportion of *S* belonging to the class *i*. The information gain represents the expected reduction in entropy based on the decrease in entropy after the dataset is split on the feature (See Eq.5).

$$Gain\ (S, A) = Entropy\ (S)\ - \sum_{v \in avlues\ (A)} \frac{|S_v|}{|S|} \cdot Entropy\ (S_v). \tag{5}$$

The feature with the highest information gain will split first. The *Gain*(*S*,*A*) of a feature *A* relative to a collection of examples *S* provides information about the target function value; given the value of some other feature *A* that splits *S* into the subsets S_v (Protić & Stanković, 2020, p.10).The characteristics of the medium DT classifier are: fast prediction speed, low memory usage, easy interpretability and medium model flexibility, i.e. medium number of leaves for finer distinctions between classes. The maximum number of splits is 20 (Protić & Stanković, 2018, p.48).

Feedforward Neural Network

The objective of the FNN is to minimize an output error in accordance with the back-propagation algorithm. The FNN transfer function used in the experiments is given with Eq. 6.

$$y_i(\mathrm{w}, \mathrm{W}) = F_i\left(\sum_{j=1}^{q} W_{ij} f_j \left(\sum_{l=1}^{m} w_{ji} x_l + w_{j0} \right) + W_{f0} \right). \tag{6}$$

where x_l are inputs, y_i are outputs, **w** and **W** are weight matrices, f_j and F_i denote the transfer functions of hidden and output layers, *m* represents the number of inputs, *q* represents the number of outputs, and w_{j0} and W_{f0} denote biases. The objective of the FNN presented in this paper is to minimize the output error in accordance with the Levenberg-Marquardt (LM) algorithm (Levenberg, 1944, pp.164-168) (Marquardt, 1963, pp.431-441).The LM algorithm performs a combined training process: around the area with complex curvature, the LM switches to the gradient descent (GD) algorithm until the local curvature is proper to make a quadratic approximation. Then it approximately becomes the Gauss-Newton(GN) algorithm which can speed up the convergence (Kwaket al, 2011, pp.327-340). The structure of the FNN presented in this paper is 9 inputs, 9 weights in the hidden layer and one output. The transfer function in the hidden layer is tangent hyperbolic while the output layer's transfer function is linear.

Experiments

A key criterion which differentiates classification techniques is prediction accuracy which represents the ratio of the number of instances correctly classified to the total number of instances (See Eq. 7).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \qquad (7)$$

where TP (true positive) represents the number of positive samples correctly predicted by the classifier, FN (false negative) represents the number of positive samples wrongly predicted as negative, FP (false positive) represents the number of negative samples wrongly predicted as positive and TN represents the number of negative samples correctly predicted by the model (Ambedkar & Kishore Babu, 2015, pp.25-29). Additionally, processing time (t), false positive rate (FPR) and false negative rate (FNR) are also measurement criteria (Nguyen & Armitage, 2008, p.56). Processing time (t) is a sum of the training and testing time. FPR represent the fraction of negative samples predicted as a positive class (See Eq. 8).

$$FPR = \frac{FP}{TN + FP}. \qquad (8)$$

FPR is a measure of accuracy for a test. It is defined as the probability of rejecting the null hypothesis, i.e. it is a probability that a false alarm will be raised; a positive result will be given when the true value is negative (Split, 2020). Ideally, FPR should be low (0.1 or less). A low FPR indicates that the classifier does not classify many irrelevant examples as relevant (Shirabad et al, 2007, p.198).

FNR represent the fraction of positive samples predicted as a negative class (Eq.9):

$$FNR = \frac{FN}{TP + FN} . \qquad (9)$$

FNR is the probability that a true positive will be missed by the classifier.

The experiments presented here are conducted on three pre-processed daily records from the Kyoto 2006+ dataset (See Table 3) and performed using Intel(R), Core(TM) i7-2620M CPU 2.7GHz processor, with 16GB RAM Installed memory.

Table 3 – Daily records – number of instances
Таблица 3 – Количество экземпляров в день
Табела 3 – Број инстанци по дану

Day	**N*umber* of instances**
03/02/2007	57278
14/02/2007	58317
27/02/2007	57278

All classifiers are trained so that 70% of daily records are used for training and 30% are used for testing. The results on accuracy, FPR, FNR, and processing time are given in Figures 1-4, respectively.

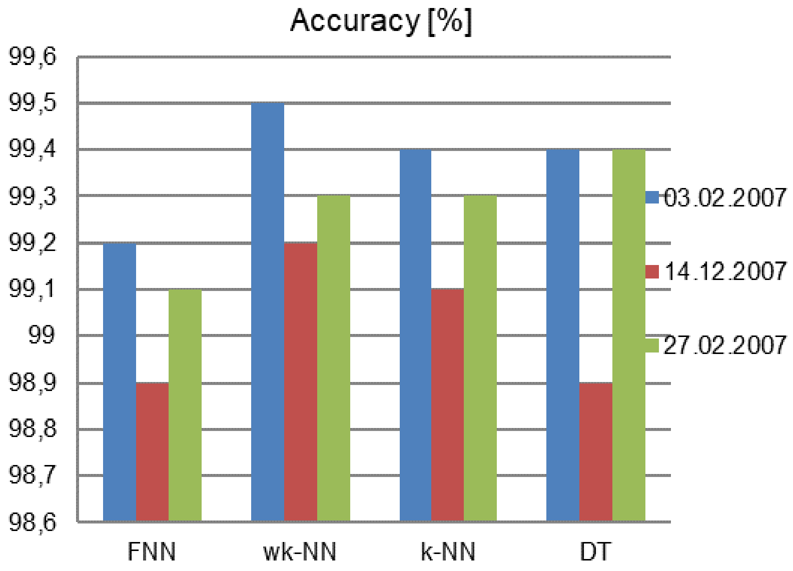

Figure 1– Accuracy
Рис. 1 – Точность
Слика 1 – Тачност

As it can be seen from Figure 1, the wk-NN has the highest accuracy of all of the models (up to 99.5%). However, the accuracies of both k-NN and DT are also high (99.4%). The FNN is less accurate than the other models but its accuracy is still very high (99.2%).

Low FPR (see Figure 2) indicates that the models classify a small number of relevant examples as irrelevant, so the probability a false alarm will be raised is very low. Although all models show a low probability of false alarm, the k-NN model classifies the highest number of irrelevant examples as relevant.

FNR has the highest value for the DT model, trained on the 14.02.2007 dataset. The lowest value of FNR gives the FNN trained on the 27.02.2007 dataset. However, FNRs of all classifiers are lower than 0.8%, and lower than 0.2% if DT is not considered as relevant.

As it is expected, the processing time is high for the lazy learners (the k-NN and the wk-NN) and exceeds 50s. The FNN and the DT show significantly shorter processing time (more than 10 times).

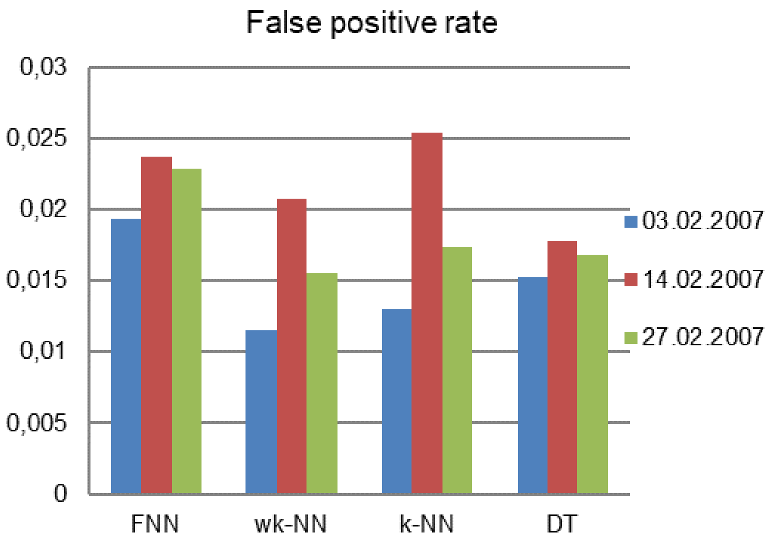

Figure 2 – False positive rate
Рис. 2 – Ложноположительный показатель
Слика 2 – Мера лажно позитивних детекција

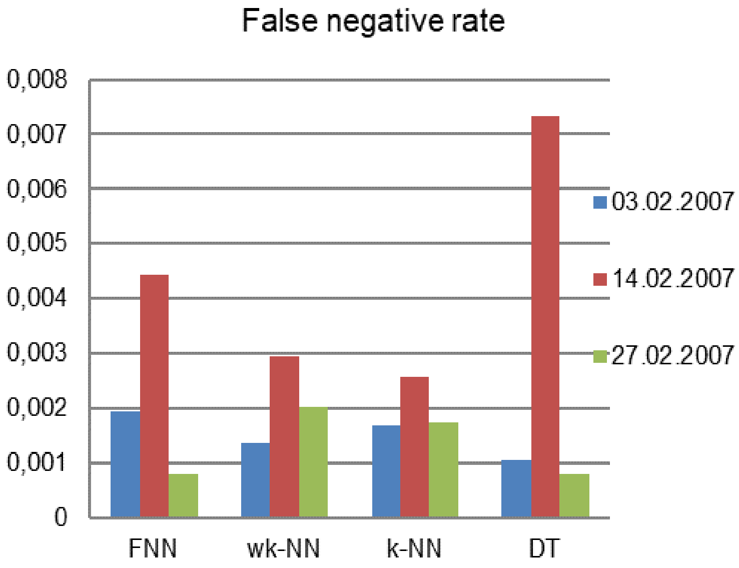

Figure 3 – False negative rate
Рис. 3 – Ложноотрицательный показатель
Слика 3 – Мера лажно негативних детекција

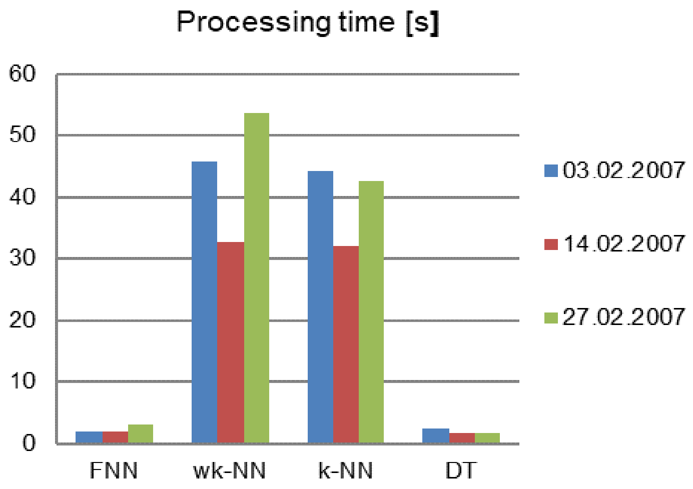

Figure 4 – Processing time
Рис. 4 – Время обработки
Слика 4 – Време обраде

Conclusion

The three-step pre-processing algorithm for feature selection and instance normalization resulted in improving performances of four binary classifiers and in decreasing processing time.

The algorithm reduced three training sets derived from the Kyoto 2006+ dataset. Accuracy, false positive rate, false negative rate, and processing time are given as measures of the performances of the classifiers.

The results show the highest accuracy of the wk-NN model. Low false positive rates indicate that the models classify a small number of relevant examples as irrelevant. FNR is significantly higher for the DT model than for FNN, k-NN, and wk-NN models.

The processing time of the lazy learners is significantly higher than the processing time of eager learners.

References

Ambedkar C. & Kishore Babu, V.2015. Detection of Probe Attacks Using Machine Learning Techniques. *International Journal of Research Studies in Computer Science and Engineering*, 2(3), pp.25-29 [online]. Available at: https://www.arcjournals.org/pdfs/ijrscse/v2-i3/7.pdf [Accessed: 29 June 2020].

Ashok Kumar, D. & Venugopalan, S.R. 2018.*A Novel algorithm for Network Anomaly Detection using Adaptive Machine Learning*. Singapore: Springer Singapore.

Kwak, Y.T., Hwang, J.W., & Yoo, C.J. 2011. A new damping strategy of Levenberg-Marquardt algorithm for multilayer perceptrons. *Neural Network World*, 21(4), pp.327-340. Available at: https://doi.org/10.14311/NNW.2011.21.020.

Levenberg, K. 1944. A method for the solution of certain problems in least squares. *Quarterly of Applied Mathematics*, 2, pp.164-168 Available at: https://doi.org/10.1090/qam/10666.

Marquardt, D.W. 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), pp.431-441 [online]. Available at: https://www.jstor.org/stable/2098941?seq=1 [Accessed: 29 June 2020].

Nguyen, T.T.T. & Armitage, G. 2008. A Survey of Techniques for Internet Traffic Classification using Machine Learning. *IEEE Communications Surveys & Tutorials*, 10(4), pp.56-76. Available at: https://doi.org/10.1109/SURV.2008.080406.

Protić, D.D. 2018. Review of KDD CUP ’99, NSL-KDD and KYOTO 2006+ Datasets. *Vojnotehnički glasnik/Military Technical Courier*, 66(3), pp.580-596. Available at: https://doi.org/10.5937/vojtehg66-16670.

Protić, D. & Stanković, M. 2018. Anomaly-Based Intrusion Detection: Feature Selection and Normalization Influence to the Machine Learning Models Accuracy. *European Journal of Formal Sciences and Engineering*, 2(3), pp.101-106. Available at: http://dx.doi.org/10.26417/ejef.v2i3.p101-106.

Protić, D. & Stanković, M. 2020. Detection of Anomalies in the Computer Network Behavior. *European Journal of Formal Sciences and Engineering*, 4(1), pp.7-13. Available at: http://dx.doi.org/10.26417/ejef.v4i1.p7-13.

Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D. & Nakao, K. 2011. Statistical Analysis of Honeypot Data and Building of Kyoto 2006+ Dataset forNIDS Evaluation. In: *Proc. 1st Work-shop on BADGES - Building Anal. Datasets and Gathering Experience Returns for Security*, Salzburg, pp.29-36, April 10-13. Available at: https://doi.org/10.1145/1978672.1978676.

-Split. 2020. *What is false positive rate?* [online]. Available at: https://www.split.io/glossary/false-positive-rate/ [Accessed: 29 June 2020].

Shirabad, J.S., Lethbridge, T.C. & Matwin, S. 2007. Modeling Relevance Relations Using Machine Learning Techniques. In: Zhang, D. & Tsai, J.J.P. (Eds.) *Advances in Machine Learning Applications in Software Engineering, Chapter VIII*, pp.168-207. Hershey, PA: Idea Group Pub. (IGI Global research collection). Available at: https://doi.org/10.4018/978-1-59140-941-1.ch008.

-Takakura. 2020. *Traffic Data from Kyoto University’s Honeypots* [online]. Available at: http://www.takakura.com/kyoto_data/ [Accessed: 29.06.2020].

Tsigkritis, T., Groumas G. & Schneider M. 2018. On the Use of k-NN in Anomaly Detection. *Journal of Information Security*, 9(1), pp.70-84. Available at: https://doi.org/10.4236/jis.2018.91006.

РОЛЬ ПРЕДВАРИТЕЛЬНОГО ПРОЦЕССИРОВАНИЯ ПРИ ОБНАРУЖЕНИИ АТАК, ОСНОВАННЫХ НА АНОМАЛИЯХ

Даниела Д. Протич

Вооруженные силы Республики Сербия, Генеральный штаб,
Управление информатики и телекоммуникаций (J-6),
Центр прикладной математики и электроники,
г. Белград, Республика Сербия

РУБРИКА ГРНТИ: 20.00.00 ИНФОРМАТИКА;
20.15.05 Информационные службы, сети, системы в целом
ВИД СТАТЬИ: оригинальная научная статья

Резюме:

Введение/цель: Система обнаружения атак, основанных на аномалиях, выявляет вторжение в компьютерную сеть, основываясь на эталонной модели, которая идентифицирует нормальное поведение компьютерной сети, детектируя аномалию. Модели машинного обучения классифицируют вторжения или злоупотребления по двум группам: нормальный трафик или аномалия. Оценка моделей классификаторов является достаточно сложным процессом, так как в сложных компьютерных сетях большое количество обучающих записей.

Методы: В данной статье представлен алгоритм выбора атрибутов, который уменьшает множество данных.

Результаты: Эксперименты проведены на множестве данных Kyoto 2006+ базы и на четырех моделях классификаторов, с помощью следующих методов: метод нейронной сети с прямой связью, метод k-ближайшего соседа, метод взвешенных k-ближайших соседей и метод дерева принятия решений. Результаты проведенных экспериментов показали высокую точность моделей.

Выводы: Трехступенчатый алгоритм предварительной обработки для выбора атрибутов и нормализации экземпляров обеспечил улучшение характеристик четырех бинарных классификаторов и сократил время обработки данных.

Ключевые слова: обнаружение аномалий, машинное обучение, KYOTO 2006+.

УТИЦАЈ ПРЕПРОЦЕСУИРАЊА НА ДЕТЕКЦИЈУ НАПАДА ЗАСНОВАНИХ НА АНОМАЛИЈАМА

Данијела Д. Протић

Војска Србије, Генералштаб, Управа за телекомуникације и информатику (J-6), Центар за примењену математику и електронику,
Београд, Република Србија

ОБЛАСТ: рачунарске науке, информационе технологије
ВРСТА ЧЛАНКА: оригинални научни чланак

Сажетак:

Увод/циљ: Систем за детекцију упада који се заснива на детекцији аномалије открива напад на рачунарску мрежу на основу референтног модела који идентификује нормално понашање рачунарске мреже и детектује аномалију. Модели машинског учења класификују упаде или злоупотребе у две групе: групу нормалног саобраћаја и групу аномалија. У сложеним рачунарским мрежама број инстанци у обучавајућем скупу може бити велики, што евалуацију модела класификатора чини тешком.

Методе: У раду је приказан алгоритам за избор атрибута који смањује величину скупа података.

Резултати: Експерименти су изведени на скупу података из Kyoto 2006+ базе и на четири модела класификатора: моделу feedforward неуронска мрежа, моделу к-најближих суседа, моделу пондерисаних к-најближих суседа и моделу стабла одлучивања. Резултати показују високу тачност модела.

Закључак: Препроцесуирање трокорачним алгоритмом за избор атрибута и нормализацију инстанци резултирало је побољшањем перформанси четири бинарна класификатора и смањило време процесуирања.

Кључне речи: детекција аномалија, машинско учење, Kyoto 2006+.

Paper received on / Дата получения работы / Датум пријема чланка: 30.06.2020.
Manuscript corrections submitted on / Дата получения исправленной версии работы/ Датум достављања исправки рукописа: 11.07.2020.
Paper accepted for publishing on / Дата окончательного согласования работы / Датум коначног прихватања чланка за објављивање: 13.07.2020.