UDC 621.565.94:004.2

MACHINE LEARNING IMPLEMENTATION FOR THE CLASSIFICATION OF ATTACKS ON WEB SYSTEMS. PART 2

K. Smirnova[1], A. Smirnov[2], V. Plotnikov[3]

[1,3]Odessa National Academy of Food Technologies, Odessa, Ukraine
ORCID: [1]0000-0002-3818-8083, [2]0000-0002-9459-6292,
E-mail: [1] smirnova.kathrin@gmail.com, [2] smyrnov.aleksandr.dev@gmail.com,[3] vmplotnik@gmail.com

DOI: 10.15673/atbp.v9i3.713

Abstract: The possibility of applying machine learning for the classification of malicious requests to a Web application is considered. This approach excludes the use of deterministic analysis systems (for example, expert systems), and is based on the application of a cascade of neural networks or perceptrons on an approximate model to the real human brain. The main idea of the work is to enable to describe complex attack vectors consisting of feature sets, abstract terms for compiling a training sample, controlling the quality of recognition and classifying each of the layers (networks) participating in the work, with the ability to adjust not the entire network, but only a small part of it, in the training of which a mistake or inaccuracy crept in. The design of the developed network can be described as a cascaded, scalable neural network.

When using neural networks to detect attacks on web systems, the issue of vectorization and normalization of features is acute. The most commonly used methods for solving these problems are not designed for the case of deliberate distortion of the signs of an attack.

The proposed approach makes it possible to obtain a neural network that has been studied in more detail by small features, and also to eliminate the normalization issues in order to avoid deliberately bypassing the intrusion detection system. By isolating one more group of neurons in the network and teaching it to samples containing various variants of circumvention of the attack classification, the developed intrusion detection system remains able to classify any types of attacks as well as their aggregates, putting forward more stringent measures to counteract attacks. This allows you to follow the life cycle of the attack in more detail: from the starting trial attack to deliberate sophisticated attempts to bypass the system and introduce more decisive measures to actively counteract the attack, eliminating the chances of a false alarm system.

Keywords: Neural network, machine learning, intrusion detection system, protection of web applications, information security.

1. Introduction

Today, the security of web applications is one of the key tasks of the information security area. Most sites available on the Internet have different vulnerabilities and are regularly exposed to various types of attacks. And if an untargeted attack can be recognized by most of the intrusion detection systems on the market, targeted attacks prepared by malicious user rather than bot attacks are difficult to recognize at times because of the impossibility of predicting all possible vectors of attacks and the tools used.

In view of the fact that the task of detecting attacks can be considered as a classification (or recognition) task, neural networks are increasingly being used to solve it. As a method for detecting malicious actions against a web system, neural networks are trained on examples of attacks of each class and, in the sequel, are used to recognize whether the observed actions belong to any attack classes. One of the problems of constructing such systems with the use of neural networks is that it is necessary to build a feature space that will allow us to separate the classes of attacks among themselves, as well as separate them from normal behavior. The second problem is the detection of attacks during their non-standard conduct, when applying an attack unknown to the neural network (a bunch of attacks), as well as deliberately "tricking" the network with a malicious user [1].

At this point in time in commercial intrusion detection systems, adaptability to unknown attacks is virtually nonexistent. And the identification of an attack happens(possible) on final stage, and not at the stage of possible prevention.

When using neural networks to recognize something, the question of vectorization and normalization is always acute.

Normalization is a set of actions that allows you to bring certain standards to the agreed norm, remove noise and highlight a certain essence, based on which the neural network will already do some conclusions.

Vectorization, in turn, is the representation of certain characteristics in the form of a vector with which the neural network itself can directly work.

2. Theoretical part

To build a classifier, it is necessary to determine which parameters influence the decision about which class belongs to a particular sample [2]. In solving this problem, the following problems can arise. First, if the number of parameters is small, then a situation may arise in which the same set of source data corresponds to examples in different classes. Then it is impossible to train the neural network, and the system will not work correctly (it is impossible to find a minimum that corresponds to such a set of initial data). A prerequisite is that the source data must be consistent. To solve this problem, it is necessary to increase the dimensionality of the feature space (the number of components of the input vector corresponding to the sample). But with the increase in the dimension of the feature space, a situation may arise where the number of examples may become insufficient for learning the network, and instead of generalizing, it will simply remember the examples from the training sample and not be able to function correctly. Thus, when determining the characteristics, it is necessary to find a compromise with their number.

Further, it is necessary to determine the way in which the input data is presented for the neural network, that is, to determine the method of rationing. Normalization is necessary, since neural networks work with data represented by numbers in the range 0..1, and the source data can have an arbitrary range or even be non-numerical data. Various methods are possible, ranging from a simple linear transformation to the required range and ending with multidimensional analysis of parameters and nonlinear normalization, depending on the effect of the parameters on each other.

If in the field of pattern recognition and text everything is conditionally simple - certain algorithms for processing natural languages are used or noise reduction algorithms are used on the image, then in the field of data protection and intrusion detection things are somewhat more complicated.

When we talk about the processing of natural languages, we are talking about the correspondence of some words to the semantic load and its emotional coloring, as well as the context. And, using knowledge of synonyms, it is possible to build a primary graph of the correspondence of words to synonyms and a second graph of contexts for complex speech turns that will solve the problem with a sufficiently high level of quality. The problems of typos can be easily solved using a Hamming correction.

The Hamming distance is the number of positions in which the corresponding symbols of two words of the same length are different. In a more general case, the Hamming distance is applied to strings of the same length of any q-ary alphabets and serves as a metric of difference (a function defining the distance in a metric space) of objects of the same dimension. Initially, the metric was formulated to determine the measure of the difference between code combinations (binary vectors) in the vector space of code sequences, in this case the Hamming distance d (x, y) between two binary sequences (vectors) x and y of length n is the number of positions in which they are different. For example, d {1011101, 1001001} = 2.

Also, the use of convolutional code can show good results for solving the problem of typs. A convolutional code is an error correcting code in which, at each cycle of the encoder's operation, the k characters of the input semi-infinite sequence are converted to n> k symbols output, and the previous m symbols also participate in the conversion; the linearity property is fulfilled (if the two encoded sequences x and y correspond to the code sequences X and Y, then the encoded sequence ax + by corresponds to aX + bY).

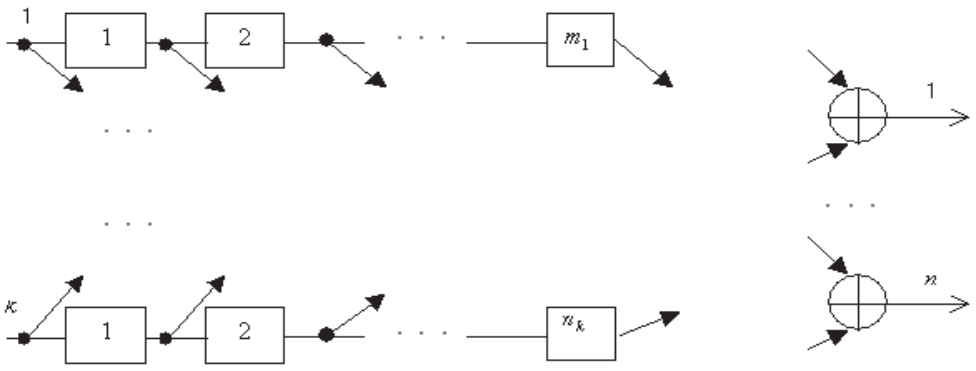

Fig. 1 – General convolutional code coding scheme

At each cycle of the encoder's operation, ‘k’ information symbols enter its input, along with the symbols stored in the shift registers, go to the inputs of those adders with which there is a connection. The result of the addition is ‘n’ code symbols ready for transmission. Then, in each shift register, a shift occurs: all cells are shifted to the right by one digit, with the

leftmost cells filled with input symbols, and the extreme right ones being erased. After that, the tact repeats. The initial state of the registers is known in advance (usually zero).

Despite the good results of these methods, they concern only the issues of recognizing texts and images outside the "aggressive" environment - in such an environment as forums or comments, or the recognition of any images on the flow. Consider the case where a person (in the context of an article an intruder) intentionally distorts and attempts to confuse a neural network, adding more and more sophisticated noises to images or words.

3. Practical part

Lets pretend that we have an attacker who attacks the system and is currently collecting information on the availability of the ability to implement SQL code for execution. It substitutes the following parameters for the query parameters - UNION SELECT @@ version. In this version, normalization is not needed, since we easily translate words into a vector, comparing them with a dictionary and writing into the vector each word that was found in its slot. The vector of signs is ready, the neural network recognizes in it an attempt to make an attack and successfully reflects it. This scheme works in theory. However, as for practical data, there are a huge number of circumvention techniques by modifying the query externally, but without changing its essence. By coding and obfuscation, for example, an attacker can take the query above and convert it, for example, to: U / ** / N / ** / I / ** / O / ** / N + S / ** / E / ** / L / ** / E / ** / C / ** / T / ** / + @ / ** / @ / ** / 0x76657273696f6e. As you can see, this text, if we make an ordinary vectorization on it, does not give anything, we get an empty vector of unknown words.

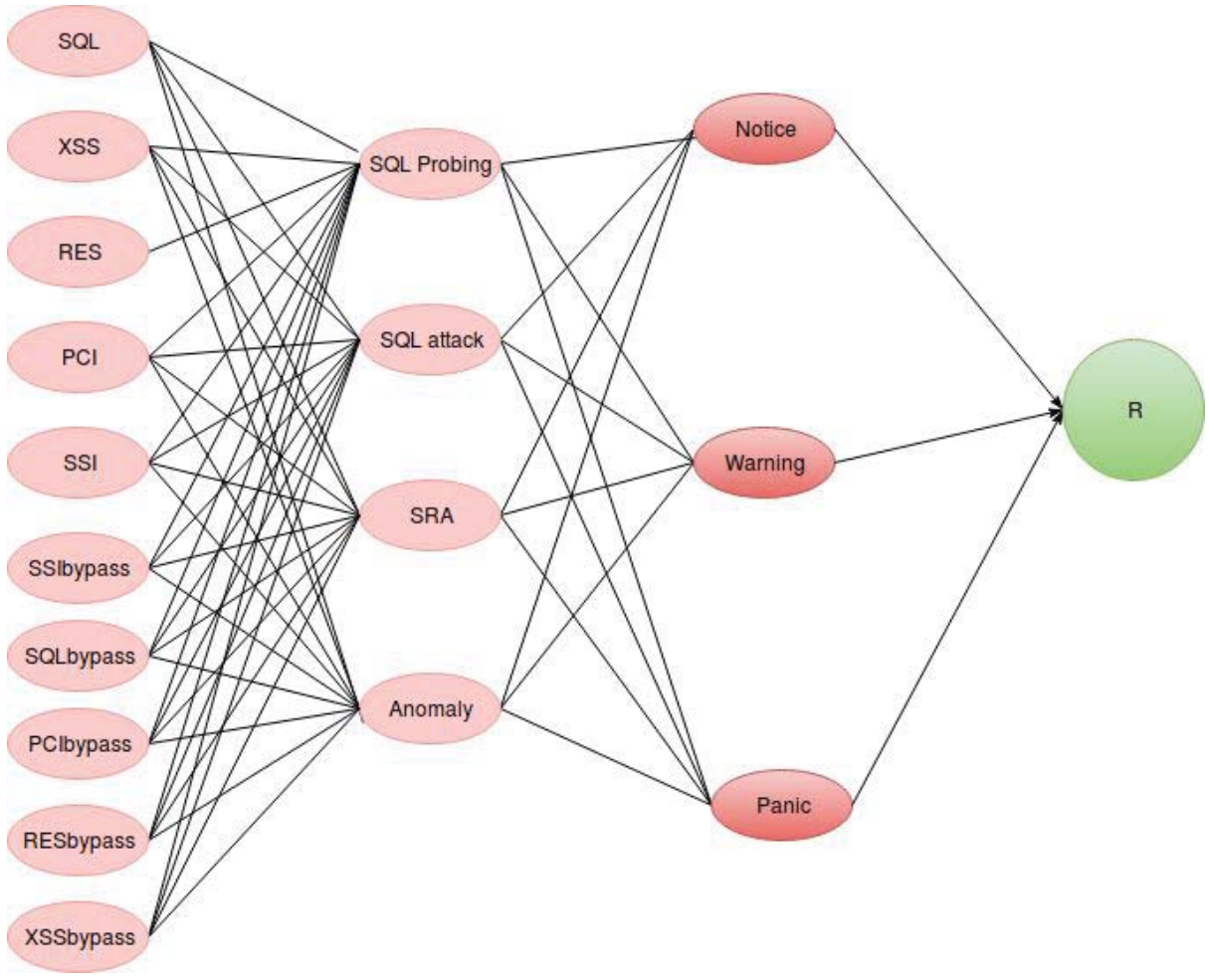

Fig. 2 – Scheme of a neural network with an additional group of neurons

The first way to solve this problem is to remove all the characters that are "noisy", and try to normalize the text. However, in practice the options to make such noisiness are hundreds, if not thousands. And if we force the system to delete all suspicious characters, we will still get a vague set at the output with a high chance. Since certain symbols in different languages mean different, we get an incorrect vector, and, consequently, the result of the network operation. For example, the symbol + in SQL and in PHP carries a different semantic load, you can not make the normalization so that the algorithm understands the nature of the query even before the work of the neural network.

The proposed solution to this problem is as follows. When learning a network, it is always necessary to form two vectors. The first vector is a vector that remains only with uppercase characters and words longer than 1. The second vector is the set of all symbols. In the first case, we not only cut out certain symbols, but also replace them with spaces, and in the second case we take into account each character that is not a letter, the digits are excluded from the analysis altogether.

With this approach, we have a more detailed neural network trained in small features. But the most important thing is that in this case it becomes possible not to solve the problems of normalization in order to avoid circumvention of the neural network and, consequently, the entire intrusion detection system. We can isolate another group of neurons on the network itself to teach it all the signs of attempts to deceive, showing it a sample of the attacks containing these same detours for each specific case.

Thus, relying on what attack vector the attacker is trying to hide, we are still able to classify any types of attacks and their aggregations as before, and also include more severe countermeasures against the attacks on the part of the user, knowing that this is not a false triggering, but a deliberate attempt to bypass the chain of actions from a simple query to a conscious attempt to deceive the intrusion detection system.

Conclusions

The most commonly used methods for normalizing and vectorizing data do not give the necessary result when used in intrusion detection systems because they are not designed to deliberately distort the signs of an attack.

The proposed solution is based on the allocation of an additional group of neurons in the first layer of the network, which are trained in all possible signs of attempts to bypass the system for all vectors of attack.

References

[1] Machine Learning Implementation For The Classification Of Attacks On Web Systems. Part 1. K. Smirnova-A. Smirnov-O. Olshevska – Avtomatyzatsiya tekhnolohichnykh i biznes-protsesiv – 2017.

[2] “Prymenenye neyronnыkh setey dlya zadach klassyfykatsyy,” *BaseGroup Labs*, 03-Sep-2015. [Online]. Available: http://www.basegroup.ru/library/analysis/neural/classification/. [Accessed: 22-Sep-2017].

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *[1301.3781] Efficient Estimation of Word Representations in Vector Space*, 07-Sep-2013. [Online]. Available: https://arxiv.org/abs/1301.3781. [Accessed: 22-Nov-2017].

[4] “Linguistic Regularities in Continuous Space Word ...” [Online]. Available: https://www.bing.com/cr?IG=1006D8A5C35A415194DEBE7B7F33A040&CID=35C1D8C68A7764E90514D3848B7165D5&rd=1&h=1Qu0WYvRDXMIuagLQw8jXrEkYbmBvC3OHhoFqgbG9Tg&v=1&r=https%3a%2f%2fwww.microsoft.com%2fen-us%2fresearch%2fwp-content%2fuploads%2f2016%2f02%2frvecs.pdf&p=DevEx,5063.1. [Accessed: 22-Nov-2017].

[5] “A Neural Network Based System for Intrusion Detection and ...” [Online]. Available: http://www.bing.com/cr?IG=E646E3819ECB40FDBAC3DCF399F2A356&CID=204ACE3E1F7567731AC5C57C1E736671&rd=1&h=gUn_pDynoUBWDwwnRLq6FHn0JGnScS60Mz3PoXxcmu8&v=1&r=http%3a%2f%2fresearch.cs.queensu.ca%2f%7emoradi%2f148-04-MM-MZ.pdf&p=DevEx,5065.1. [Accessed: 22-Nov-2017].

[6] “Outside the Closed World: On Using Machine Learning For ...” [Online]. Available: http://www.bing.com/cr?IG=C7A840A14BB04B4587E5FF73F1CBCE46&CID=1D514BD09BB76BC0011540929AB16A7B&rd=1&h=sv5NtZNzG8MREQB41ZtwuqqA0Zr7wlU_GyYCpbUlSSM&v=1&r=http%3a%2f%2fwww.utdallas.edu%2f%7emuratk%2fcourses%2fdmsec_files%2foakland10-ml.pdf&p=DevEx,5063.1. [Accessed: 22-Nov-2017].

Література

[1] Smirnova K., Smirnov A., Olshevska O. MACHINE LEARNING IMPLEMENTATION FOR THE CLASSIFICATION OF ATTACKS ON WEB SYSTEMS. PART 1 //Автоматизація технологічних та бізнес-процесів. – 2017. – Т. 9. – №. 2.

[2] Стариков А. Применение нейронных сетей для задач классификации //Режим доступа: – (http://www. basegroup. ru/library/analysis/neural/classification/). – 1995.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space // In Proceedings of Workshop at ICLR, 2013.

[4] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations // In Proceedings of NAACL HLT, 2013.

[5] Moradi M., Zulkernine M. A neural network based system for intrusion detection and classification of attacks //Proceedings of the IEEE International Conference on Advances in Intelligent Systems-Theory and Applications. – 2004. – С. 15–18.

[6] Sommer R., Paxson V. Outside the closed world: On using machine learning for network intrusion detection //Security and Privacy (SP), 2010 IEEE Symposium on. – IEEE, 2010. – С. 305–316.