Computer Science and Information Systems 15(1):139–162 https://doi.org/10.2298/CSIS160905039P

Evaluation of Takagi-Sugeno-Kang Fuzzy Method in Entropy-based Detection of DDoS attacks

Miodrag Petkovic, Ilija Basicevic, Dragan Kukolj, and Miroslav Popovic

University of Novi Sad, 21000 Novi Sad, Serbia
mio@eunet.rs, ilibas@uns.ac.rs,
dragan.kukolj@rt-rk.com, miroslav.popovic@rt-rk.uns.ac.rs

Abstract. The detection of distributed denial of service (DDoS) attacks based on internet traffic anomalies is a method which is general in nature and can detect unknown or zero-day attacks. One of the statistical characteristics used for this purpose is network traffic entropy: a sudden change in entropy may indicate a DDoS attack. However, this approach often gives false positives, and this is the main obstacle to its wider deployment within network security equipment. In this paper, we propose a new, two-step method for detection of DDoS attacks. This method combines the approaches of network traffic entropy and the Takagi-Sugeno-Kang fuzzy system. In the first step, the detection process calculates the entropy distribution of the network packets. In the second step, the Takagi-Sugeno-Kang fuzzy system (TSK-FS) method is applied to these entropy values. The performance of the TSK-FS method is compared with that of the typically used approach, in which cumulative sum (CUSUM) change point detection is applied directly to entropy time series. The results show that the TSK-FS DDoS detector reaches enhanced sensitivity and robustness in the detection process, achieving a high true-positive detection rate and a very low false-positive rate. As it is based on entropy, this combined method retains its generality and is capable of detecting various types of attack.

Keywords: Network security; Fuzzy neural networks; Distributed denial of service attacks; Intrusion detection; Takagi-Sugeno-Kang model

1. Introduction

Denials of service (DoS) attacks are continuous cause of financial and reputational damage. Any organization that relies on the internet for communication with customers may be a victim of this type of attack. The detection of attacks in source networks and the mitigation of attacks in target networks are issues that remain to be solved, since the characteristics of attacks often change, becoming more sophisticated and powerful.

The concept of using entropy for network attack detection is not new. The use of fuzzy and neural network methods, and particularly TSK, represents the next wave in this field. The novel approach used in this paper is a combination of entropy-based and Takagi-Sugeno-Kang fuzzy neural network-based methods in detecting DoS attacks. In the majority of related work, only one of these two approaches is used. The reason for the usefulness of entropy in DoS detection lies in the fact that the entropy of normal network traffic varies within a narrow band [1]; many anomalies caused by DoS attacks change the distribution of addresses and ports [2], as well as other traffic characteristics.

The strength of entropy-based detection arises from its generality [1-3]. The application of another processing level based on fuzzy logic and neural networks results in better detection characteristics. This requires an additional offline learning process using input data sets with known attack times. These learning data sets can be obtained from real sources by sniffing network traffic or from the simulated network environment. In this work, we used the ns-2 simulator, which is often used in research in this area. Our goal was to show that a combination of entropy-based and TSK-FS-based detection of DoS attacks gives rise to a robust general method with a lower false-positive rate (FPR), a higher true-positive rate (TPR) and a performance which comes close to that of specialized methods.

The following is a short review of some related work in this area. Nychis et al. [4] researched the use of entropy in the detection of many different types of attack. These authors found a correlation between certain distributions, such as address and port distributions. Based on this, they suggest the choice of a complementary traffic distribution to increase detection rate. They also found that entropy is not efficient for the detection of certain types of attacks, such as port scanning or low-intensity DDoS. Our findings confirm those in this previous work [5] and indicate that address distributions can give a satisfactory detection rate for low-intensity DDoS. In [6], the Shannon entropy detector and the chi-square detector are evaluated. Test data are obtained from public datasets, with attack periods inserted. These attack periods are very long in comparison with the short attack periods used in our experiments. A simple threshold is used for change point detection, and the rate of detection is high. These authors propose a method for an immediate response to detected attacks. In [7], the authors use sliding time windows, as used in our research, and create a traffic profile based on minimum and maximum entropy values for each window. In addition to Shannon entropy, the use of other entropy measures has also been explored. Speidel et al. [1] report the application of T-entropy; these authors conclude that smaller and shorter events can be more easily detected by means of T-entropy than by Kolmogorov complexity estimation. In [8-9], the authors compare a Tsallis-based and a Shannon-based detector. In [8], DDoS traffic was injected into Abilene and Geant data sets used in [2,10] and the performances of the two approaches were measured. In [11], a discrete wavelet transformation is used for detection of traffic anomalies. The transformation of the normal traffic is treated as noise, while the pulses are treated as anomalies that should be preserved and detected. The proposed method, referred to as an anti-denoising method, results in a reduced FPR. In our paper, the TSK-FS method plays the same role in the reduction of noise and FPR.

As for fuzzy-based methods, in [12], the Takagi-Sugeno fuzzy system is used with the predefined data set KDDCup99 rather than network simulation; this is based on connection attributes rather than on entropy. Experiments also show a high detection rate and a low rate of false alarms. In [13], a DDoS fuzzy detector is constructed based on the mean packet inter-arrival times, and the detection rate is empirically evaluated as over 80%. In many cases, however, it is difficult to distinguish legitimate behavior such as flash crowds, or a surge in traffic to a particular target, from DDoS attacks. This is investigated in [14]; it constitutes a limitation on all DoS detection methods.

2. Denial of Service Attacks

Denial of service (DoS) attacks affect the availability of network resources and cause significant damage to business organizations and government agencies every year. The intention of an attacker is to prevent legitimate users from using the attacked service. During DoS attacks, attackers hit their target with a large amount of requests or data, exhausting its resources and preventing legitimate users from obtaining access. Large servers are usually robust enough to defend against attacks originating from a single machine, and thus DoS attacks are often carried out in the form of distributed attacks (DDoS) from a large number of single machines. These machines are under the control of the attacker and form a so-called 'botnet'. They are infected with malicious software and thus can be controlled by the attacker, who starts a DDoS and in fact never accesses the target. Botnets are often rented as DDoS-for-hire services in the IT underground. Very often, DDoS attacks are reflected: the attacker uses a spoofed source address for a target and sends a broadcast message to an amplifying network. The entire amplifying network then responds to the forged source address, i.e., to the address of the target, causing an enormous number of requests. Examples of reflected attacks are Smurf, which uses a broadcast ping, and Fraggle, which sends a UDP echo to the broadcast address. Another well-known and still largely used form of attack is the SYN flood, in which the attacker, a TCP client, initiates a large number of three-way TCP handshakes without the intention of ever completing them. The first packet in the handshaking process is SYN. The server then enters the SYN-RECEIVED state, allocates a memory block to process it, and creates a half-open connection. The aim is to exhaust the number of allocated memory blocks, preventing further TCP connections and thus denying the service to legitimate clients. A detailed classification of DoS attacks can be found in [15].

Distributed denial of service is still a growing problem. According to a Kaspersky Lab report for Q3 2016 [16], the average number of attacks per day in Q3 was about 600, with a peak figure of 1746. The majority of DDoS attacks last up to 4 hours, although the duration of the longest reported attack was more than 7.6 days. By type, the most frequently used is still SYN-DOS (81%), followed by TCP-DOS (8.20%), and HTTP-DoS (7.56%). The most powerful attack was 620Gbit/s at its peak. Such attacks pose a threat not only to specific web resources but also to the data centers and even to the infrastructure of internet service providers.

3. TSK Fuzzy Detector Synthesis

3.1. Entropy in DDoS Detection

In information theory, entropy is used as a measure of the unpredictability or uncertainty of a system. Entropy is highest for truly random data from an information source, and is lowest when an information source gives completely predictable data. The concept of entropy is derived from the field of thermodynamics. Although at a practical level the connection between informational entropy and thermodynamic

entropy is not evident, a similar entropy equation is used in information theory as a measure of the information of a single random variable which is the output of a discrete information source. This is expressed by the widely used Shannon's equation, which is also used in this work:

$$H(Z) = -\sum_{i=1}^{n} p(z_i)\log(p(z_i)) \quad (1)$$

where z_i is an instance of Z and $p(z_i)$ is the probability that Z takes the values of z_i.

In the process of detecting DDoS attacks, entropy is computed for a sample of consecutive packets for chosen header fields. A comparison of the value of entropy for the chosen header fields in one sample to the entropy of the corresponding fields of another sample provides a mechanism for detecting changes in randomness. It has been observed [3] that while a network is in a normal state, the entropy values for various header fields fall within a narrow range. When the network is under attack, these entropy values change significantly and can be detected.

The Shannon entropy is not the only measure of information uncertainty. In this research, we also used the parameterized Tsallis entropy:

$$H(Z) = \frac{1-\sum_{i=1}^{N} p_i^q}{q-1} \quad (2)$$

Tsallis entropy converges to Shannon entropy when the parameter q tends to 1. When the value of parameter $q > 1$, high-probability events have a higher contribution to the resulting entropy, while for $q < 1$, events with low frequency are the main contributors. Thus, by fine-tuning q it is possible to change the sensitivity of event detection.

Measures of information complexity are also often used in similar studies. Kolmogorov complexity is in general not computable, and only its estimation can be used [17]. T-entropy has also been proposed as a good estimation of information complexity [1] and this has close correspondence with known physical entropies. T-complexity is a measure which expresses complexity as the number of steps required to build a string. In an effort to acquire a Shannon entropy compatible information measure from T-complexity, Titchener [18] thus proposed the inverse logarithmic integral $li^{-1}(x)$ as a suitable function for linearising T-complexity. Finally, the T-entropy of a string x is defined as the gradient of the T-information of x with respect to the length |x| of x:

$$H_T(x) = \frac{dI_T(x)}{d\,|x|} \quad (3)$$

Since these methods use recursive string parsing to calculate complexity estimation, they require more computational effort than the calculation of entropies using Equations (1) and (2).

The strength of entropy-based anomaly detection lies in its generality. A significant change in entropy level may be a sign that a network is under attack, regardless of the type of attack. As such entropy based methods belong to wider class of anomaly-based detection methods which detect deviation from 'normal' traffic. Thus, entropy-based

methods are capable of detecting zero-day attacks. On the other side, there are signature-based methods, which are crafted for the detection of a specific type of attack, are in most cases incapable of detecting other types of attack. Although in this research we use specific attack types as SYN flood, the proposed method is general in nature and not dependant on any specific attack type.

3.2. TSK Fuzzy Neural Network Synthesis

The Takagi-Sugeno-Kang (TSK) model is characterized by a high accuracy of modeling combined with a very fast learning process. The TSK model was proposed as a systematic approach to generate fuzzy rules from a given set of input-output data and where the structure of the system is not known in advance. The detection of anomalies in network traffic using large amounts of input-output data requires this type of model. This offers an advantage over the widely used Mamdani fuzzy model, which is more intuitive and, as such, is more suitable for human input. TSK model gives a more compact and computationally efficient representation than the Mamdani system, and allows for adaptation techniques so that membership functions can be customized. This is very important in modeling highly dynamic internet traffic. Furthermore, the number of rules can be much smaller in this approach than in the Mamdani fuzzy model is applied, even for complex systems, as described in [19]. The TSK model has already been successfully applied to a number of real-world problems such as the approximation of a static non-linear function, stock market predictions, predictions of natural gas consumption, estimation of DC motor speed [20] and TCP throughput control [21], to mention only a few.

The idea of the TSK model is that a complex system can be presented as a combination of inter-linked subsystems. These subsystems can be described with simpler functional dependencies. If the dependence is considered to be linear, and if one rule corresponds to exactly one subsystem, the final model with C rules can be represented in the following form:

$$R_i: \text{If } \boldsymbol{x_1} \text{ is } \boldsymbol{A_{i1}} \text{ and } x_2 \text{ is } \boldsymbol{A_{i2}} \text{ and } \ldots \text{ and } \boldsymbol{x_n} \text{ is } \boldsymbol{A_{in}} \quad \text{then } y_i = \boldsymbol{a_i x} + b_i,\ i = 1, 2, \ldots, C \tag{4}$$

where R_i is the i-th rule; x_1, x_2, … x_n are inputs; A_{i1}, A_{i2}, …, A_{in} are fuzzy sets assigned to each input variable; and y_i is the output variable of the i-th rule. Vector $\boldsymbol{a}_i$ and scalar b_i are parameters of the consequent linear function. In our case, the inputs are entropy values from within the observed time window, and the outputs are the values of the current DDoS attack.

The output of the TSK fuzzy model for an input of $\boldsymbol{x}_k$ is:

$$\hat{y}_k = \sum_{i=1}^{C} [\omega_i(x_k) y_i(x_k)],\ k = 1, 2, \ldots,\ N \tag{5}$$

where $\omega_i(x_k)$ is the normalized activation level for the i-th rule of the k-th input sample, and is given by:

$$\omega_i(x_k) = \frac{\beta_i(x_k)}{\sum \beta_j(x_k)} \tag{6}$$

where β_i is the firing strength of the *i*-th rule [20]. The input-output data space is partitioned in clusters, and the algorithm uses training data with N input-output samples:

$$z_k = [x_k^T; y_k]^T \; k = 1,..., N \tag{7}$$

where *N* is the total number of all samples.

The dimension of the input data is *N* and the dimension of the output is one. Each cluster represents a certain subsystem in which input-output data values are concentrated.

The data from the learning set are divided into the obtained clusters and then interpreted as rules as in Equation (4). A single-layer neural network is generated, as shown in Fig. 1. Each node in the output layer is actually the center of the corresponding cluster.

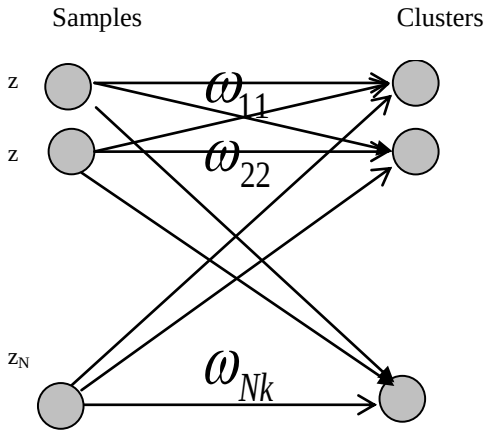

Fig. 1. Structure of the single-layer neural network

The number of clusters *C* is fixed and is a parameter of the algorithm. A detailed description of the clustering algorithm is given in [22]. The input samples from the test dataset, i.e., the samples for which we want to estimate outputs, are then assigned to clusters on the basis of their proximity to the center of the cluster. The Euclidean distance between the input sample *x* and the center of each cluster is calculated. When the smallest distance is found, the input sample *x* is assigned to the nearest cluster A_i. The output vector *y* is then calculated using Equation (4). A more detailed mathematical derivation of the parameters $\boldsymbol{a}_i$ and b_i is omitted here and can be found in [20].

4. TSK-FS Detector Implementation

4.1. Entropy Calculation

The following algorithm used is our entropy-based detector implementation. The distribution of the selected variable is monitored during small subintervals of 0.1 seconds. There is a sliding window of 10 successive subintervals. For each subinterval, the algorithm calculates the value of the monitored variable distribution. For every monitored variable, an array of subintervals is allocated. In this research, we monitor the distribution of the transferred bytes and packets, as well as the different source and destination addresses during each subinterval. At the end, the entropy values are calculated for each subinterval.

For Shannon and Tsallis entropy, equations (1) and (2) are used. For T-entropy, mapping from addresses to symbols should be applied. Every 5 bits of IP address represent one symbol, starting with 'A' for binary 00000b. The strings built that way are input in libflott [23] program, proposed in [24], which outputs T-entropy.

Network traffic could be taken from various sources, for example from real networks using sniffing tools. However, the generation of DDoS attacks, which can disturb normal business activities, is a problem here.

There are also publicly available traffic data sets, both with and without included DDoS attacks. The DDoS attacks in these data sets are fixed and as such not configurable. In addition, these data sets often do not have time-labeled attacks, meaning that the output of the detection process cannot be accurately verified.

Third option is the use of a network simulator. The drawback of this method is that it is very difficult to achieve a realistic simulation. However, the main advantage of network simulators is their configurability, which is valuable for research purposes. For this reason, we have chosen to use the ns-2 network simulator in this work. Furthermore, ns-2 is open source software, and we were thus able to change the source code to reflect specific requirements. A comparison of various simulators [25], including the new generation product ns-3, indicates that ns-2 still satisfies the needs of this research. To verify the proposed method, we used the public CAIDA and DARPA data sets in the final experiments.

4.2. Learning Process

For the purposes of the learning process, the learning data are treated as containing known attack values at each point, i.e. within each subinterval. The attack variables can take two possible values: a nonzero value if the attack is under way, and zero if there is no attack at this point in time. There is no exact value for the attack variable; however, in this research we found that the optimal value is of the same magnitude as the mean entropy. Fig. 2 presents the software components of this system.

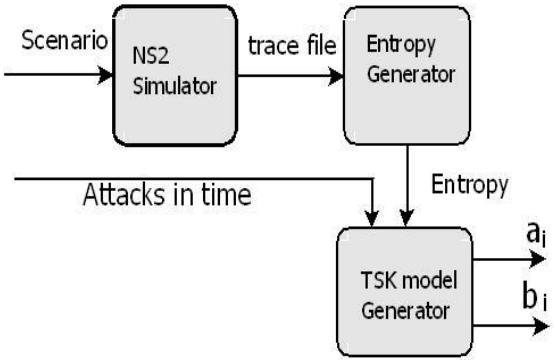

Fig. 2. The learning process

The output from the ns-2 simulator is a standard trace file. In the next step, entropy values are calculated for each 0.1 s subinterval, as described previously in Section 4.1.

The output from the entropy generator, which also forms the input for the TSK-FS model generator, is an array of entropy values for a single monitored variable (e.g., destination address) in 0.1 s subintervals and the values of the known output, i.e., the value of the attack for each corresponding entropy value.

A generator input vector is created in the TSK-FS model. The input vector consists of 10 consecutive entropy values from the subintervals within a single sliding window, plus the attack value for the time of the most recent subinterval. Consecutive values are needed in order to incorporate the entropy trend into the model and to suppress noise. At the next step, N (the number of samples) input vectors are processed in an offline TSK-FS learning process. In this case, we used an input value of C=5 clusters, which we found optimal for performance. Any further increase in C does not sufficiently improve detection to justify the higher computational effort. At the end of this process, parameters $\boldsymbol{a}_i$ and b_i for each rule ($i = 1\ldots C$; C=5) of Equation (3) are calculated.

4.3. Detection Process

The detection process is shown in Fig. 3. The test data are in the same format as the learning data but without the attack values, which are yet to be estimated. The array of attack values $\hat{y}_i$ is calculated by the TSK detector using Equation (4). The first two software components are the same as in the learning process. The third component is the TSK-FS detector. The detector performs the following steps. For each input sample, the Euclidean distance to all clusters is calculated. For the nearest cluster, an appropriate fuzzy rule is applied using Equation (4), and output $\hat{y}_i$ is obtained.

The automatic detection of a change point of $\hat{y}_i$, i.e., an attack in time, is performed using a simple cumulative sum control chart (CUSUM) (see [26]). The mean value is estimated using an exponential weighted moving average (EWMA) method, where weighting factors decrease exponentially [26]. It is assumed that the outputs $\hat{y}_i$ are independent and identically distributed values. There are two hypotheses: distribution before the changes and distribution after the changes. In the parametric version, the test of the change is based on the log-likelihood ratio; in the non-parametric version of

CUSUM this is based on a custom function. In this work, we used the non-parametric version.

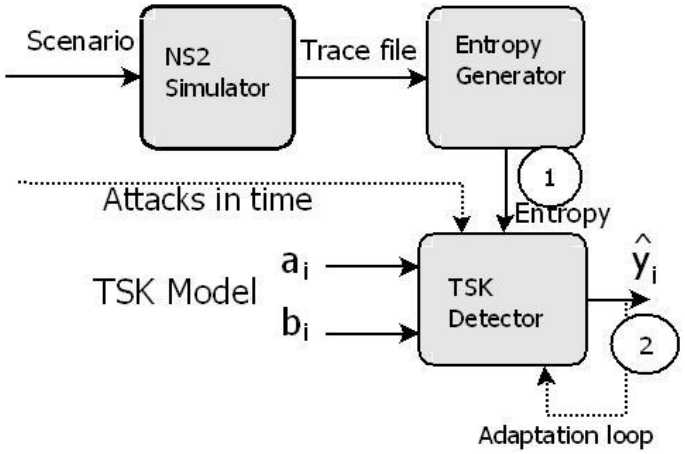

Fig. 3. The detection process

For the detection, the following equations were used:

$$\mu_n = \beta_1 y_n + (1-\beta_1)\mu_{n-1} \quad (8)$$

$$d_n = \max\{0, d_{n-1} + y_n - (\mu_n + K)\},\ d_0 = 0 \quad (9)$$

$$\sigma_n^2 = \beta_2 \sigma_{n-1}^2 + (1-\beta_2)(y_n - \mu_n)^2,\ H = h\sigma_n \quad (10)$$

H is a decision threshold and depends on standard deviation σ_n. If $d_n > H$, a change is detected. The values for h, the decision factor and K, the allowance factor, depend on the level of entropy; these vary from 2.0 to 6.0 and from 0.01 to 0.05 respectively in our experiments. The detection strongly depends on the variations in these values, and this fact is considered in this research. μ_n is an estimation of the mean value of output series y_n. The counter d_n accumulates the deviations of y_n from μ_n that are greater than the allowance factor K (minimum deviation). β_1 and β_2 are EWMA adaptation factors, and their values are fixed to 0.75 and 0.90 in this research. σ_n is the standard deviation of output series y_n.

Finding optimal values for the allowance factor K and threshold factor h may be a challenge. These values affect the robustness and sensitivity of the detection, and values which lead to high detection characteristics in one experimental setup could cause a poor detection rate in another. For this reason, we added adaptation capabilities to the detector implementation. These are represented by the dotted lines in Fig. 3. The known attacks from the learning dataset and the feedback from output y are used to find the local maximum of the difference (TPR − FPR). The adaptation process starts from a point (h_0, K_0) and finds the maximal (TPR – FPR) value in its vicinity. The new point

(h_1, K_1) becomes the starting point for the next iteration step. At the end of the process, the obtained values for h and K are used for further measurements.

5. Simulation and Results

Experiments were carried out using two different topologies (Figs. 4 and 5) and different types of traffic. For each topology, one learning dataset was used to generate the TSK-FS model, and several test datasets were used to measure detection rates for various levels of attacks, starting from a very low level of attack.

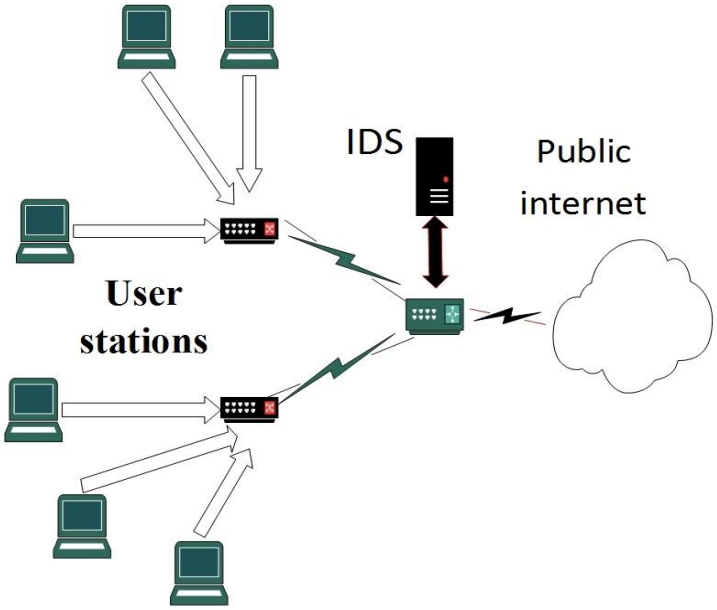

Fig. 4. Edge topology. The attacks originate from user stations in local network, while the targets are within the public internet. The detection point is at the edge/output of the monitored network

The learning data are generated using 15 attack points and has a total duration of 75 seconds. The exact times of the beginning and end of each attack are known in advance. The number of attacks in the test datasets is also 15. Each attacker-controlled station performs a SYN flood attack every five seconds. The duration of a single attack is a random variable with a normal distribution, a mean value of one second and a standard deviation of 200 ms. The experiments were carried out using the ns-2 simulator package, version 2.35, and the simulated attack was a SYN flood DDoS. The original ns2 source code of TCPAgent component was modified to perform the simulation of the SYN attack response.

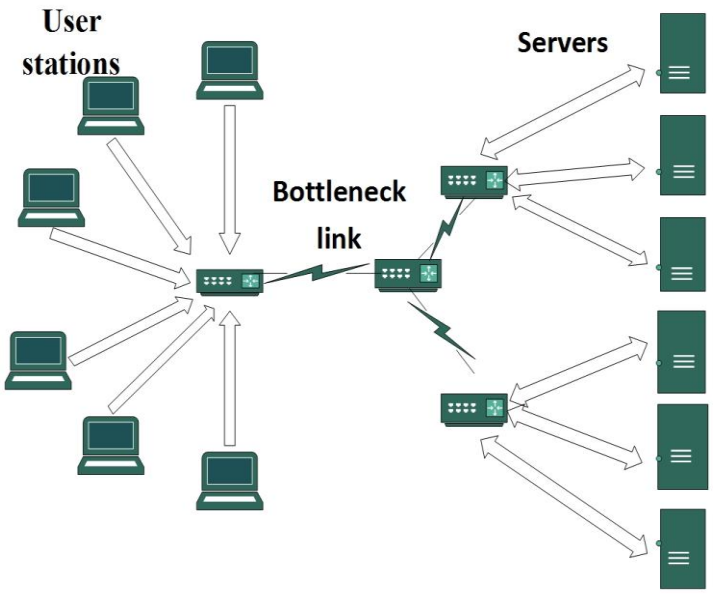

Fig. 5. Large-scale topology. The attacks start from user stations within the public internet, while the targets are local servers

A series of experiments was carried out with the number of attacks varying from 20 to 150 for the large-scale topology and from 20 to 80 for the edge topology. For each experiment, CUSUM change point detection was applied to two points in the detection process (points 1 and 2 in Fig. 3). The first point is at the output of entropy calculation and the second point is at the output from TSK-FS. The goal of the experiments was to show that applying the TSK-FS method increases the detection rate and suppresses false positives.

The edge network topology used in the simulation contains a local network with a base station serving 250 user stations; of these, there are between 20 and 80 attacker-controlled stations and one attack target, as shown in Fig. 4. The intrusion detection system (IDS) sensor is positioned at the gateway, monitoring the traffic to and from the public internet. The host, which is the target of the DDoS attack, is within the public internet, and thus outside of the monitored network. The attack is detected by monitoring the outbound traffic. This topology enables the detection of DDoS attacks near their source, at a point where they can be mitigated locally.

The baseline traffic was generated using a set of 100 constant bit rate (CBR) agents and 100 agents which generated HTTP traffic. Each CBR transfer contained an object with a mean size of 10k bytes, while each HTTP agent contained an object with a mean size of 30kB. The time interval between file transfers was exponentially distributed, with a mean value of 30 s. Fig. 6 presents the entropy values and the corresponding outputs from the TSK-FS detector for a simulation scenario using 30 attackers. The upper signal is the destination address entropy, while the lower signal represents the output y of the TSK-FS detector. The attacks take place between 250 and 325s. For each experiment, CUSUM change point detection was applied on both signals, as defined by Equations (8), (9), and (10).

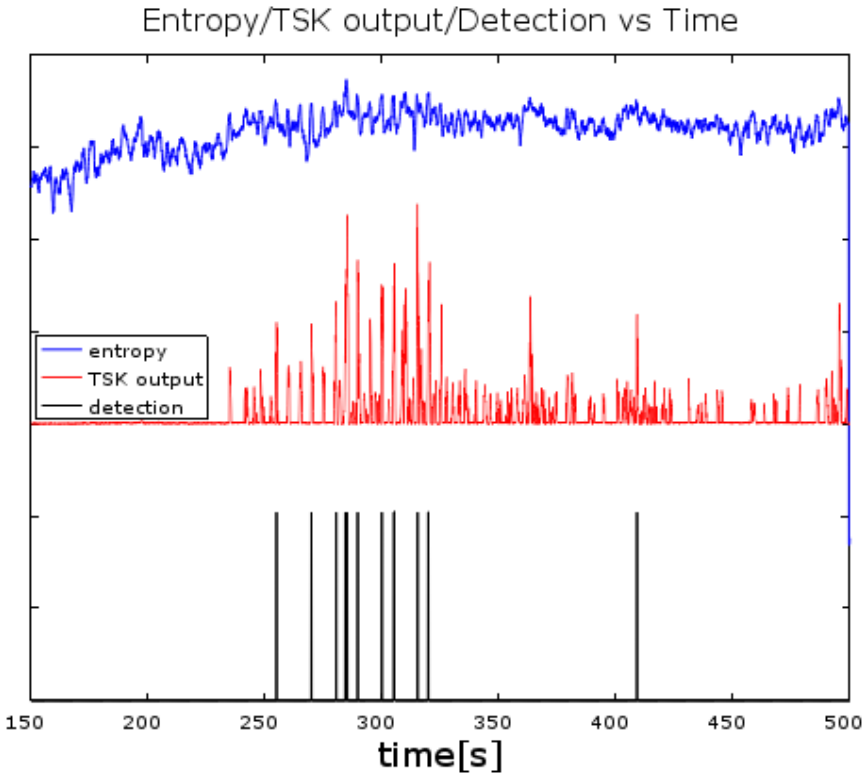

Fig. 6. Stages in detection: entropy (upper signal), TSK-FS output (middle signal) and final detection (lower signal). Attacks take place between 250 and 325s. Attack strength is low, but detection is still effective.

In general, the cause of false positive detections in entropy-based methods is the noise in the entropy values. Fig. 6 demonstrates the key concept of this research, which is that the application of the TSK-FS filter in entropy processing suppresses the noise and emphasizes impulsive changes, reducing the number of false positives and enhancing true positives. The upper signal represents the entropy of the network traffic, while the lower signal shows the detected attacks in time. The example given in Fig. 6 shows the case when attacks are very low and shows how TSK-FS method is capable to filter out even very small changes in entropy time series.

Fig. 7 presents the dependency of detection rates on threshold value h for a fixed allowance factor K. The diagram is centered on the optimal h and K values where difference between TPR and FPR reaches a local maximum. The reliability of the detection is measured in terms of the closeness of TPR to 100% and FPR to 0%. The figure shows that the application of TSK-FS improves the quality of detection.

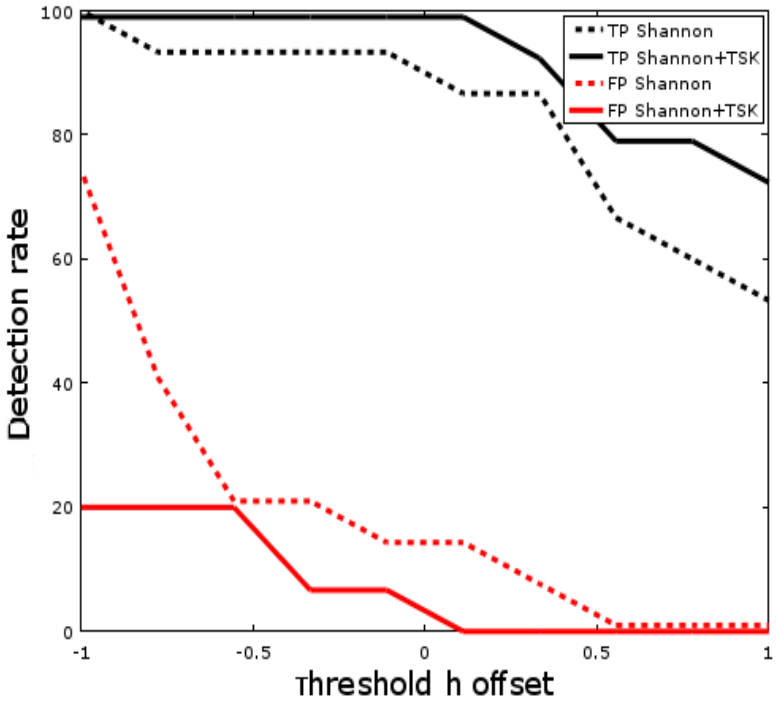

Fig. 7. Detection rate against threshold for edge topology, using Shannon entropy with CUSUM change detection for the optimal value of the allowance factor *K*

In our research, it was found that the allowance factor *K* in CUSUM change point detection should not take a fixed value; both *h* and *K* must vary in order to find the best detection rate. Fig. 8 illustrates the three-dimensional dependency of detection rates on both *h* and *K* for Shannon entropy, while Fig. 9 presents the same dependency when TSK-FS is applied to Shannon entropy. The detection is more reliable if the TPR and FPR surfaces are closer to 100% and 0% respectively. Point (0, 0) represents the point of optimal values of *h* and *K*, where the distance between TPR and FPR is a maximum. Hence, the *h* and *K* axes on the diagram are actually offsets from the optimal values for the threshold and allowance factors. Detection rates on vertical axis are percentages of detected attacks out of total number of attacks

From the diagrams it can be seen that the detection is more robust, and the distance between TPR and FPR remains high for a wider range of values of *h* and *K*, compared to the method without the use of TSK-FS processing.

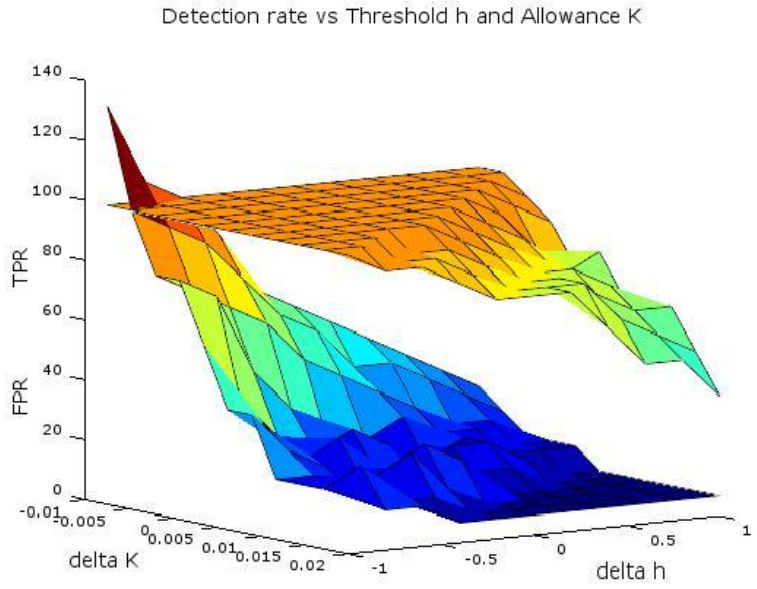

Fig. 8. True (upper surface) and false (lower surface) detection rates against two CUSUM parameters *h* and *K* for the edge topology. Shannon's entropy is used without TSK-FS

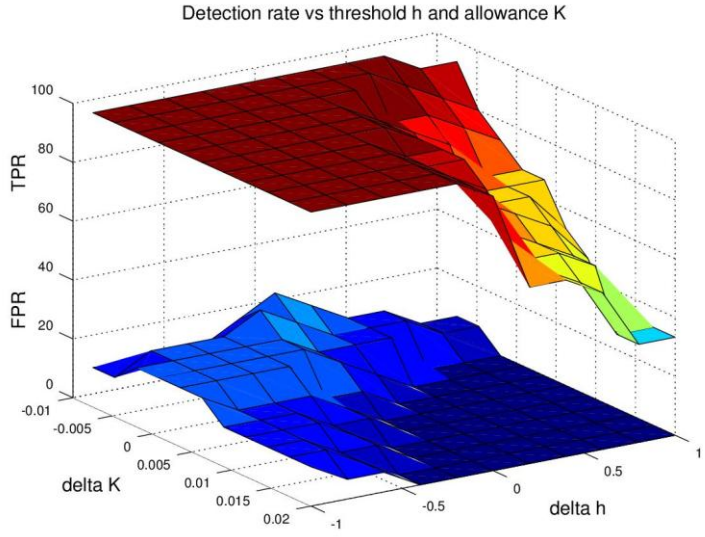

Fig. 9. True (upper surface) and false (lower surface) detection rates for the edge topology and Shannon's entropy with TSK-FS applied. The CUSUM method is used for the final change point detection

Table 1 presents the results for the destination address distribution for the edge network topology using Shannon, Tsallis and T-entropy, when the attack strength changes from high to low. The TPR and FPR values are presented only for the optimal choices of *h* and *K*. The optimal values for the threshold and allowance parameters are obtained from the optional adaptation loop (the dotted lines in Fig. 3).

It can be seen that the true-positive rate is higher, and FPR is lower, when TSK-FS is applied. Even in the case of very low-level attacks, when normal variations in entropy are of the same magnitude as the variations caused by attacks (last row in Table 1 for each entropy type), TSK-FS still detects some attacks, whereas the detection based solely on Shannon entropy does not detect any attack. Similar experiment in [5], table I show similar results for similar simulation setup with Shannon entropy. Tsallis entropy gives a slightly better detection rate, especially for a low-level attack and when combined with TSK-FS. T-entropy gives poor detection rate, although TSK-FS also reduces FPR. This is especially true for low-level attacks where detection is even not possible. This may be explained with high sensitivity of T-entropy, which detects pattern changes in signals, and as such produces more noise than Shannon and Tsallis entropy. The problem of noise in application of T-entropy is discussed in Eimann's thesis [27].

Table 1. Comparison of detection rates for Shannon, Tsallis and T-entropy, with and without the applied TSK-FS method for the edge network topology and optimal parameter choices. The number of attackers (first column) varies from 20 to 80

Number of attackers	**TPR**	**FPR**	**h**	**K**	**TPR**	**FPR**	**h**	**K**
	Shannon				**Shannon+TSK-FS**			
80	94%	6%	6.8	0.02	100%	0%	4.0	0.03
60	87%	0%	3.4	0.07	100%	6%	4.6	0.01
40	20%	0%	6.6	.025	94%	6%	4.3	0.005
20	0%	0%	-		46%	0%	2.8	0.035
	Tsallis				**Tsallis+TSK-FS**			
80	100%	6%	5.8	0.03	100%	0%	4.2	0.03
60	87%	13%	5.6	0.03	100%	0%	4.6	0.01
40	61%	6%	5.0	0.005	87%	0%	3.8	0.005
20	26%	13%	8.8	0.03	54%	0%	3.3	0.03
	T-entropy				**T-entropy+TSK-FS**			
80	80%	40%	5.2	0.025	56%	6%	4.8	0.025
60	46%	26%	4.8	0.03	33%	0%	4.6	0.020

From Figs. 8 and 9 it is obvious that the detection rate is more robust when TSK-FS is applied, for a deviation of the threshold *h* and allowance *K* from the optimal values. This is especially true for low values of the allowance *K*, which actually represents the sensitivity of detection. For low values of *K*, entropy-only detection shows a significant rise in false positives (Fig. 8). The false positive rate is significantly lower for the TSK-FS method, and this does not increase even for low threshold *h*.

In the second set of scenarios, the simulated topology is a network composed of 470 user stations in the public domain. The number of attacker-controlled stations varies

from 20 to 150. There are 40 servers in the local network, of which one is the attack target. The topology is an unbalanced dumbbell topology, illustrated in Fig. 5. The attack target is part of the server tree. This topology simulates large-scale attacks from the internet on a server within a local network. There is a link node between the client and the server parts of the network, and this node contains an IDS detector. The attack is detected by monitoring the inbound traffic. All clients are grouped into one cluster, and the baseline traffic is HTTP sessions generated by the clients. The duration of simulation is 500 s. The simulated attack varies from the low-intensity type, in which only 20 stations are controlled by the attacker, to 150 attacker-controlled stations. The number of sessions is 200, and each session contains 250 pages with a single object. The object size is the Pareto II variable with an average value of 120 and shape parameter 1.2. The attacks are in range of 250–325 s while the network load is high. The dataset from the scenario with 60 attacker stations was used for the learning phase of TSK-FS.

Figures 10 and 11 present the true and false detection rates versus threshold factor *h* and allowance factor *K* for one chosen experiment with 110 attackers and Tsallis entropy. Detection rates on vertical axis are percentages of detected attacks out of total number of attacks.

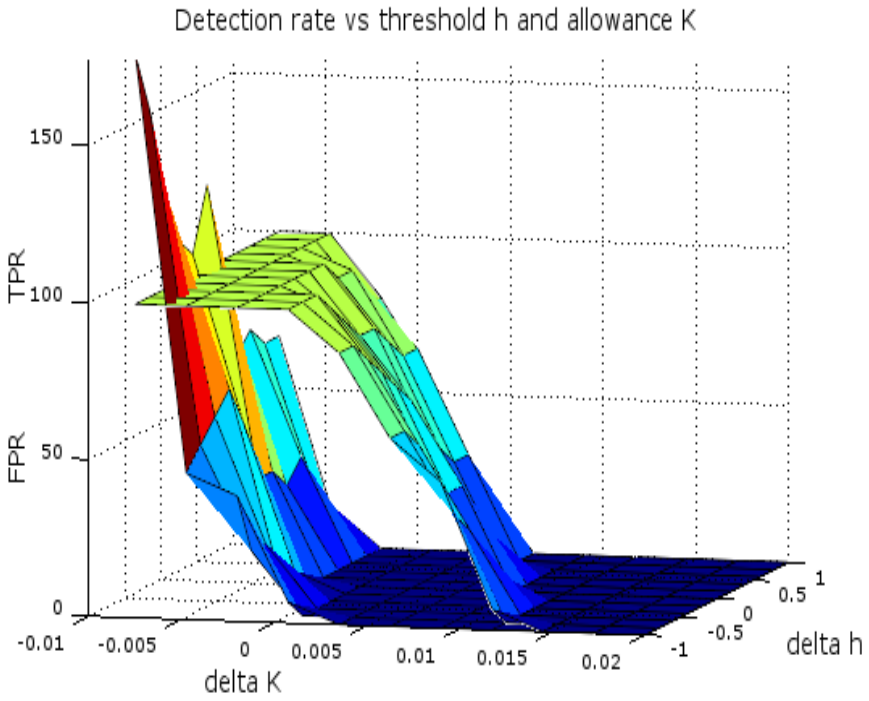

Fig. 10. True (upper surface) and false (lower surface) detection rates against CUSUM parameters *h* and *K* for the large-scale topology and Tsallis entropy without TSK-FS

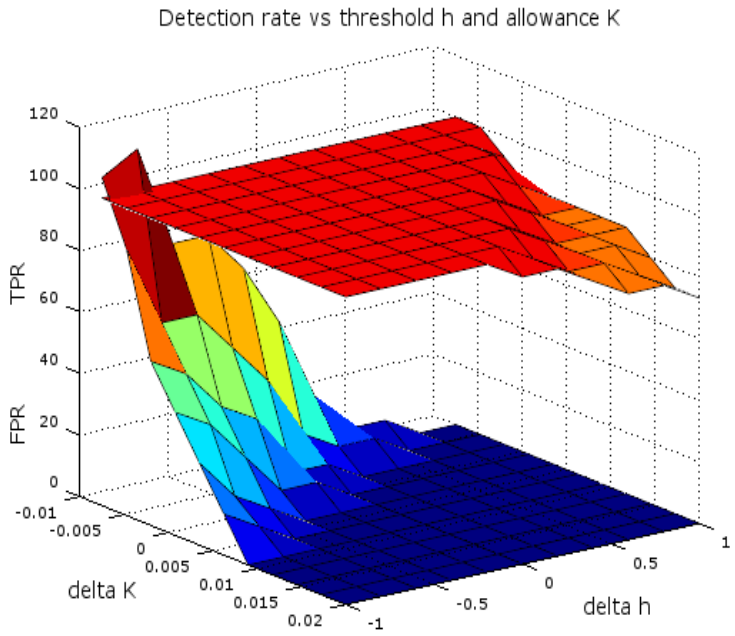

Fig. 11. True (upper surface) and false (lower surface) detection rates against CUSUM parameters *h* and *K* for the large-scale topology and Tsallis entropy with TSK-FS

In Fig. 10 we can see that the surfaces representing TPR and FPR are close to each other i.e. detection is less robust. In a similar way to the experiments for the edge topology, detection is only possible for very narrow range of allowance factor *K*. The corresponding diagram for the same experiment using TSK-FS detection is presented in Fig. 11. We can see that the detection rate is more robust when the TSK-FS filter is applied, and the differences between TPR and FPR are high for a wide range of CUSUM change point detection parameters *h* and *K*. This result is important for the possible practical implementation of TSK-FS DDoS detection in network equipment, since, once determined, the factors *h* and *K* may give a stable quality of detection when the environment changes dynamically.

Table 2 presents the results for the destination address distribution for the large-scale topology when the attack strength changes from low to high. The results in Table 2 are presented only for the optimal values of *h* and *K* for given attack strengths. The same observation applies to this set of experiments as for the edge network topology. Detection with applied TSK-FS outperforms detection based solely on entropy. Even for low-strength attacks, the TSK-FS method still detects an attack at the point where entropy variations caused by DDoS attacks are too low to be detected by simple entropy-based detection. Again, the parameterized Tsallis entropy gives slightly better results than Shannon entropy, while T-entropy gives acceptable results only in the case of strong attacks. In all experiments FPR for the TSK-FS method remains low. Similar results are presented in [9], table 1. Our results in experiments without TSK show less FPR for similar setup, but this is because in [9] factor K is fixed, while in this work it varies, so the local maximum of (TPR – FPR) is higher. In [7] authors performed experiments for few entropy types. Although experimental setup is different, TPR and

FPR are close to our results without TSK. The same experiments were carried out for other distributions. The experiments show similar results for source address distribution for both topologies. Byte and packet distribution show poor performance with both methods for the topologies used.

Table 2. Comparison of detection rates for Shannon, Tsallis, and T-entropy with and without applied TSK-FS method for the large-scale network topology and optimal parameter choices. Number of attackers varies from 20 to 150

Number of attackers	TPR	FPR	h	K	TPR	FPR	h	K
	Shannon				**Shannon+TSK-FS**			
150	80%	13%	1.7	0.01	100%	6%	4.3	0.02
110	94%	6%	1.6	0.01	100%	0%	3.0	0.03
70	74%	20%	1.4	0.01	87%	6%	5.7	0.01
50	47%	13%	1.6	0.01	74%	0%	5.4	0.01
30	13%	13%	1.7	0.01	26%	0%	4.5	0.025
20	0%	20%	1.7	0.01	20%	6%	3.9	0.025
	Tsallis				**Tsallis+TSK-FS**			
150	100%	0%	4.8	0.025	100%	0%	6.0	0.035
110	100%	0%	4.8	0.015	100%	0%	6.0	0.035
70	100%	6%	3.8	0.015	100%	0%	5.8	0.035
50	94%	20%	3.8	0.012	86%	13%	5.6	0.035
30	13%	0%	3.8	0.015	47%	13%	4.6	0.03
20	0%	0%	4.8	0.012	23%	6%	6.0	0.025
	T-entropy				**T-entropy+TSK-FS**			
150	87%	20%	4.6	0.01	67%	0%	3.9	0.01
110	6%	6%	5.0	0.01	33%	0%	4.2	0.015
70	0%	20%	5.2	0.015	20%	6%	4.2	0.01

A cross-topology experiment was also carried out, i.e., the learning dataset was taken from the large-scale topology and the generated model was then applied to the test data taken from the edge network topology. The results are presented in Table 3. The detection rates are slightly lower than for the model obtained from the appropriate topology, but low FPR is still shown for both destination and source address distributions. A cross-entropy experiment, i.e., when the TSK-FS model generated for Shannon entropy was applied to data obtained by applying Tsallis entropy, did not show improved detection rate, although detection was still possible.

Table 3. Cross-topology detection results. The learning dataset and the test dataset are from different network topologies

Attack strength	TSK-FS method			
	TPR	**FPR**	**h**	**K**
80 attackers	87%	0%	5.8	0.05
60 attackers	73%	6%	9.5	0.055
40 attackers	54%	6%	6.5	0.15
20 attackers	33%	13%	8.1	0.075

In the final set of experiments, available public data sets have been used to verify the method. The model from the large-scale topology was applied to the CAIDA data sets [28]. The purpose of these experiments was not to accurately test detection rate but to apply the TSK-FS method to real traffic as a step towards its practical implementation. Fig. 12 presents entropy values and corresponding outputs *y* from the TSK-FS detector for data set 20070804_141436 and for a destination address distribution. There are no false positives, and the period during which the network was under attack is detected as three attacks taking place very close in time, which is the correct detection from a practical point of view.

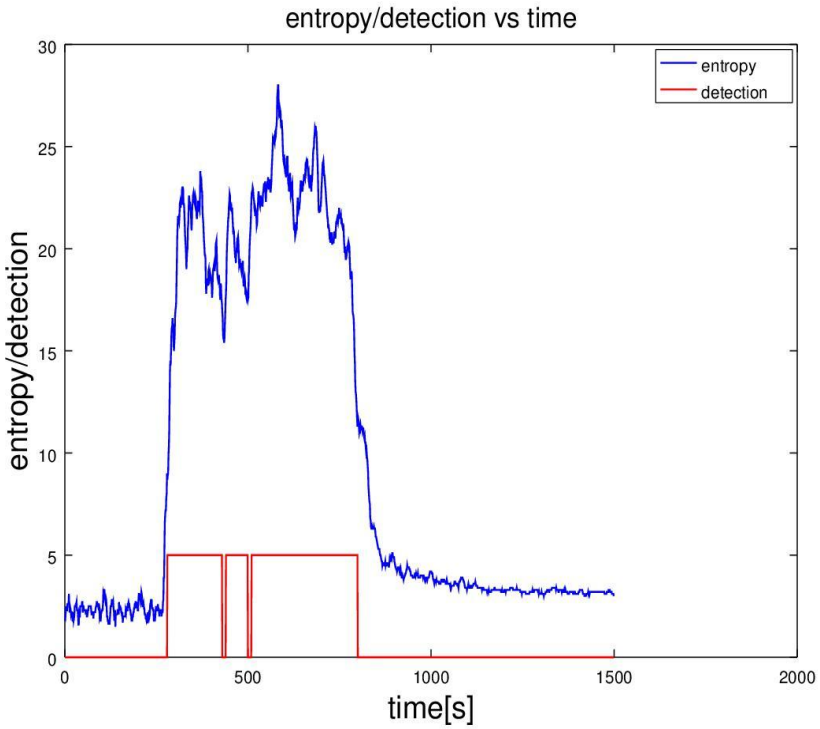

Fig. 12. Entropy and TSK-FS output for the sample CAIDA data set

In the second experiment with publicly available data sets, presented in Fig. 13, the data set is DARPA_2009_DDoS_attack-20091105 [29]. The victim target with address 172.28.4.7 is outside of the network, similar to edge topology example. So, the model from edge topology is used for this data set. In opposite to previous example, DDoS attack consists of very frequent single attacks which sometimes last as short as few microseconds. From practical point of view it is important to detect the attacks as a whole, from its starting moment to the end, and not necessary every single flow. More than 55% of short attacks are detected, spanning across the whole DDoS attack duration, which makes the detection successful.

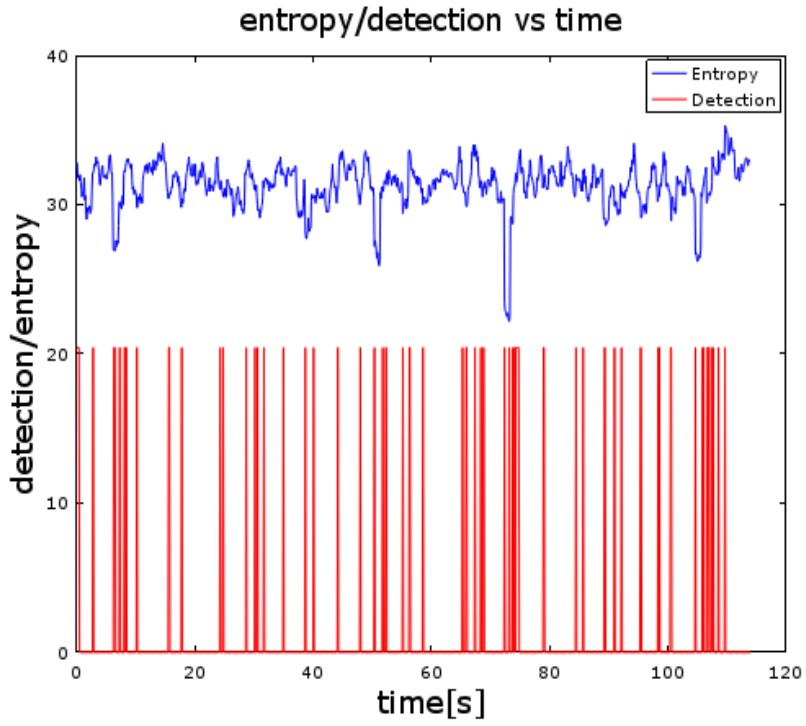

Fig. 13. Entropy and TSK-FS output for the sample DARPA data set

6. Conclusions

This paper presents an evaluation of a combined method for the detection of outbound DDoS attacks based on entropy with the Takagi-Sugeno-Kang fuzzy neural network detector. The CUSUM method is used for final change point detection in all experiments. The monitored TCP packet distributions are the source and destination addresses and the number of bytes and packets. For entropy calculation, Shannon's, Tsallis and T-entropy equation are used. The trace files from the ns-2 network simulator are used as learning and test data. Experiments were performed on two topologies and three types of entropy: a local network topology with constant bit rate baseline traffic, with a detector on its edge; and a large-scale topology with HTTP baseline traffic. Experiments were carried out using three different entropies: Shannon's, Tsallis and T-entropy.

The experimental evaluations confirm that the method significantly increases the DDoS detection rate, and is more robust when the configuration changes, in comparison with the method using direct detection on entropy values. By applying the TSK-FS method on any three types of entropy, the noise in the entropy time series is suppressed, enabling more accurate change point detection. The experiments show that slightly better results are achieved if Tsallis entropy is used and poorer results for T-entropy. False positives are significantly reduced, even in the case of low intensity attacks. The

true-positives detection rate remains high and the false-positive detection rate remains low for a wide range of values of configuration parameters. This finding is important for practical uses of this method in IDS equipment. The method was also successful when a TSK-FS model generated on one specific network topology was applied to a different network topology. Finally, the method was successfully applied on real-world public data sets.

This method is effective for the entropy of destination and source addresses distributions, although it has poor performance when applied to byte and packet entropy distributions. As an entropy-based method, it retains its generality and is capable of detecting attacks regardless of their type. On the other hand, the fuzzy and neural network approach increases the sensitivity and robustness of detection. Additional robustness and better network event coverage could be achieved if this method is combined with a set of specific signature-based methods.

There are several directions for possible future research. One is the fine-tuning of the algorithm to provide a lower number of false positives when dealing with flash crowds, that is, a burst of legitimate traffic to single destination. Another direction is to determine the optimal set of input variables for the TSK-FS model to achieve a better detection rate in a dynamic environment. The derivation of entropy, the values of some packet header fields, and packet inter-arrival times are good candidates for combining with entropy values.

Acknowledgements. This work has been partially supported by the Ministry of Education and Science of the Republic of Serbia under the Project TR32030.

Anonimized attack traffic CAIDA ”DDoS Attack 2007”, from 2007-08-04 to 2007-08-04, has been used in this work. Provided by Center for Applied Data Analysis, http://www.caida.org

Anonimized attack traffic DARPA_2009_DdoS_attack-20091105, from 2009-11-04 to 2009-11-04, has been used in this work. Provided by USC/LANDER, https://ant.isi.edu/datasets

Program https://github.com/ardeego/libflott, provided by Niko Rebenich and Stephen Neville, University of Victoria, British Columbia, Canada, has been used for T-entropy calculation

References

1. Speidel, U., Eimann, R., Brownlee, N.: Detecting network events via T-entropy. 6th International Conference on Information, Communications & Signal Processing ICICS, Singapore, pp. 1–5, 2007. doi: 10.1109/ICICS.2007.4449642 (2007)
2. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications SIGCOMM 05, Philadelphia, vol 31, issue 4, pp. 217–228, 2005. doi:10.1145/1080091.1080118 (2005)
3. Wagner, A., Plattner, B.: Entropy based worm and anomaly detection in fast IP networks. 14th IEEE International Workshops on Enabling Technologies: Infrastructure For Collaborative Enterprise, Linkoping, pp. 172–177, 2005.doi: 10.1109/WETICE.2005.35 (2005)
4. Nychis, G., Sekar, V., Andersen, D.G., Kim, H., Zhang, H.: An empirical evaluation of entropy-based traffic anomaly detection. Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement, Vouliagmeni, Greece, pp. 151–156, 2008, doi:10.1145/1452520.1452539 (2008)

5. Basicevic, I., Ocovaj, S., Popovic, M.: Evaluation of entropy-based detection of outbound denial-of-service attacks in edge networks. Security and Communication Networks; vol. 8, issue 5, pp. 837-844, 2015, doi: 10.1002/sec.1040 (2015)
6. Feinstein, L., Schnackenberg, D., Balupari, R., Kindred, D.: Statistical Approaches to DDoS Attack Detection and Response. Proceedings of the DARPA Information Survivability Conference and Exposition, vol. 1, pp. 303-314, 2003.doi: 10.1109/DISCEX.2003.1194894 (2003)
7. Berezinski, P., Jasiul, B., Szpyrka, M.: An Entropy Based Network Anomaly Detection Method. Entropy 2015, vol. 17, issue 4, pp. 2367-2408, 2015. doi:10.3390/e17042367 (2015)
8. Ziviani, A., Gomes, A.T.A., Monsores, M.L., Rodrigues, P.S.S.: Network anomaly detection using nonextensive entropy. IEEE Communications Letters, vol. 11, issue 12, pp. 1034–1036, 2007. doi: 10.1109/LCOMM.2007.070761 (2007)
9. Basicevic, I., Ocovaj, S., Popovic, M.: Use of Tsallis entropy in detection of SYN flood DoS attacks, Security and Communication Networks, vol. 8, issue 18, pp. 3634-3640 ,2015. doi: 10.1002/sec.1286 (2015)
10. Uhlig, S., Quoitin, B., Balon S., Lepropre J.: Providing public intradomain traffic matrices to the research community. ACM SIGCOMM Computer Communication Review, vol. 36 issue 1, pp. 83-86, 2006. doi:10.1145/1111322.1111341 (2006)
11. Vancea, F.: Intrusion detection in NEAR system by Anti-denoising Traffic Data Series using Discrete Wavelet Transform, Advances in Electrical and Computer Engineering, vol 14, issue 4, pp.43-48, 2014, doi: 10.4316/AECE.2014.04007 (2014)
12. He, Liang: An Improved Intrusion Detection based on Neural Network and Fuzzy Algorithm. JOURNAL OF NETWORKS, vol. 9, no. 5, 2014. doi:10.4304/jnw.9.5.1274-1280 (2014)
13. Shiaeles, S.N., Katos, V., Karakos, A.S., Papadopoulos, B.K.: Real time DDoS detection using fuzzy estimators. Computers & Security, vol. 31 issue 6, pp. 782-790, 2012. doi:10.1016/j.cose.2012.06.002 (2012)
14. Handley, M., Rescorla, E.: DoS considerations. RFC 4732, RFC Editor, (2006)
15. Mirkovic, J., Reiher, P.: taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communications Review; vol. 34, issue 2, pp. 39–53, 2004. doi: 10.1145/997150.997156 (2004)
16. Kaspersky Lab: Kaspersky DDoS Intelligence Report Q3 2016, Tech. Rep. October 2016. (2016)
17. Kulkarni A., Bush, S.: Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics. Journal of Network and Systems Management, vol. 14, issue1, pp. 69-80, 2006. doi: 10.1007/s10922-005-9016-3 (2006)
18. Titchener, M.R.: A Deterministic Theory of Complexity, Information, and Entropy. In Proceedings of IEEE Information Technology Workshop, February 1980
19. Ke, Z., Nai-Yao, Z., Wen-Li, X.: A Comparative Study on Sufficient Conditions for Takagi–Sugeno Fuzzy Systems as Universal Approximators. IEEE Transactions on Fuzzy Systems vol. 8, issue 6, pp. 773-780, 2000. doi: 10.1109/91.890337 (2000)
20. Kukolj, D.: Design of adaptive Takagi-Sugeno-Kang fuzzy model. Applied Soft Computing, vol. 2, issue 2, pp. 89-103, 2002. doi:10.1016/S1568-4946(02)00032-7 (2002)
21. Bašičević, I., Kukolj, D., Popović, M.: On the application of fuzzy-based flow control approach to High Altitude Platform communications., Applied Intelligence, Springer; vol. 34, issue 2, pp. 199-210, 2011. doi:10.1007/s10489-009-0190-y (2011)
22. Kukolj, D., Atlagic, B., Petrov, M.: Unlabeled data clustering using a re-organizing neural network Cybern Syst Int J, vol. 37, issue 7, pp. 779–790, 2006. doi:10.1080/01969720600887152 (2006)
23. https://github.com/ardeego/libflott

24. Rebenich, N.: Fast Low Memory T-Transform: string complexity in linear time and space with applications to Android app store security. PhD thesis, University of Victoria, British Columbia, Canada.(2012)
25. Weingartner, E., vom Lehn, H., Wehrle K.: A Performance Comparison of Recent Network Simulators. ICC '09. IEEE International Conference on Communications, 2009. doi: 10.1109/ICC.2009.5198657 (2009)
26. Siris, V.A., Papagalou, F.: Application of anomaly detection algorithms for detecting SYN flooding attacks. Computer Communications, vol. 29, issue 9, pp. 1433–1442, 2006. doi: 10.1016/j.comcom.2005.09.008 (2006)
27. Eimann, Raimund E.A: Network event detection with entropy measures, PhD thesis, University of Auckland, New Zealand (2008)
28. Hick, P., Aben, E., Claffy, K., Polterock, J.: The CAIDA "DDoS Attack 2007" Dataset, http://www.caida.org/data/passive/ddos-20070804_dataset.xml (2007)
29. The ANT Lab: Analysis of Network Traffic, https://ant.isi.edu/datasets

Miodrag Petkovic received Dipl. Eng., and M.Sc degrees from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 1988, and 1999, respectively. He was a teaching assistant at the Faculty of Technical Sciences, University of Novi Sad. He is the author of a few distributed solutions in the area of lottery and finance. Currently he works as a professional software engineer. His research interests are in the areas of Internet protocols, OS security, network security, and computer simulation. He has authored or co-authored more than 15 scientific papers.

Ilija Basicevic received Dipl. Eng. M.Sc, and PhD degrees from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 1998, 2001, and 2009, respectively. Currently he is Associate Professor at the University of Novi Sad, teaching courses on computer networks. His research interests are in the areas of Internet protocols, Internet traffic and network security. He has authored or co-authored more than 45 scientific papers and one textbook.

Dragan Kukolj received his Diploma degree in control engineering in 1982, MSc degree in computer engineering in 1988, and PhD degree in control engineering in 1993, all from the University of Novi Sad, Serbia. He is currently a Professor of computer-based systems with Department of Computing and Control, Faculty of Engineering, University of Novi Sad. His main research interests include soft computing, data mining techniques and computer-based systems integration with applications in video processing and digital signal processing. He has published about 190 papers in referred journals and conference proceedings, 5 books and he is co-inventor of 12 national patents and patent applications and the founder and coordinator of Intellectual Property Center of University of Novi Sad.

Miroslav Popovic received his Dipl. Eng., M.Sc., and Ph.D. degrees from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 1984, 1988 and 1990, respectively. He is a Full Professor at the University of Novi Sad from 2002. Currently he is giving courses on software tools and real-time systems programming, as well as on inter-computer communications and computer networks. In the past, he has supervised many real-world projects for the industry, such as telephone exchanges and call centers

for telecommunication networks. His research interests are engineering of computer-based systems, distributed systems, and security. He has authored or co-authored about 20 peer-reviewed journal papers, more than 120 conference papers and technical reports, and the book Communication protocol engineering (CRC Press, 2006).

Received: September 5, 2016; Accepted: August 20, 2017.