2012 Third International Conference on Intelligent Systems Modelling and Simulation

Fuzzy Detection of Malicious Attacks on Web Applications based on Hidden Markov Model Ensemble

Mohammad Geraily
Computer Engineering Department
Islamic Azad University, Mashhad Branch
m.geraily@gmail.com

Majid Vafaei Jahan
Computer Engineering Department
Islamic Azad University, Mashhad Branch
vafaeijahan@mshdiau.ac.ir

Abstract—This paper represents a system, which detects malicious HTTP request and obtains the lowest false-positive rate with high detection rate. For this purpose, each extracted feature of a HTTP request is modeled by multiple hidden Markov models as a classifier ensemble. HMMs outputs of an ensemble are fused to product a probabilistic value that showing normalcy of corresponding feature. In this system, instead of a threshold, a fuzzy inference is applied to produce a flexible decision boundary. So, fuzzy sets and rules of decision module are formed manually; next, output of each HMM ensemble is converted to a fuzzy value with respect to fuzzy sets. Finally, a fuzzy inference engine uses these values to produce output that indicates whether the HTTP request is normal or abnormal. Experiments show that this approach is flexible and has acceptable accuracy in detecting requests close to the decision boundary, and false-positive rate is 0.79%.

Keywords: Hidden Markov model ensemble; Fuzzy inference; Multiple classifier System; Fusion; Soft boundary; Detection rate; False positive rate

I. INTRODUCTION

Today, the internet as one of the best human's inventions has a great influence on people's daily life and has been developed in most of the social, scientific and other aspects. In this regard, there have been some problems which one of the most important is illegal access to information. One of those that have been the most welcomed are web applications, which have been extended in communication, science, medicine, business and public services. Web applications are also not secluded from this problem and even for their structure are more attacked and have been influenced by illegal access [2]. Therefore, here, a method is presented for detecting attacks on web applications.

Intrusion detection is a supervision process on occurred events in a computer system or network which its aim is to detect symptoms based on an unauthorized access. Intrusion-detection systems are divided into two main signature based and anomaly based categories with respect to applied methods and techniques. The purpose of signature based intrusion-detection systems is to detect attacks that happened in the past; and although, they perform better than an anomaly based detection systems according to error rate, they are weak in detecting zero-day attacks, which this disadvantage has been removed in anomaly based systems. The important aim in this work is to develop an intrusion-detection system with high accuracy and the minimum error which by supervising HTTP requests and detecting an anomaly in them, detects attacks on web applications. The proposed method has been configured with respect to an anomaly based systems and has used hidden Markov model and fuzzy inference to solve an anomaly detection problem. Two important tasks in anomaly diagnosing are modeling and classifying patterns and decision which if they are done well, better performance could be observed in anomaly detecting. The proposed system has two approaches that cause mentioned tasks to be performed well and consequently, the accuracy of attack's detection to be increased. The first system's approach is to use a multiple classifier for modeling every extracted feature from an HTTP request such that by fusion obtained outputs from each classifier, it can neglect error resulting from some classifiers by the majority's vote. The second approach is to apply soft and flexible boundary for discriminating normal and abnormal HTTP requests. In the most previous works, a threshold value was used to separate them, which had a weakness in separating requests near the decision boundary. If normal or abnormal HTTP requests are obviously recognizable to each other, intrusion-detection system could act with higher accuracy. In this method, to define a proper boundary between normal and abnormal patterns, fuzzy inference has been utilized. According to two aforementioned approaches, the proposed intrusion-detection system, in addition to having acceptable detection rate, has the lowest false-positive rate. Therefore, in section 2, some works and attempts which have been done in this area will be studied. Section 3 will discuss methods of decreasing error rate and increasing accuracy of attack diagnosis on web applications. In section 4, the proposed system will be described, and it will be evaluated and tested in section 5. At last, a conclusion will be stated in section 6.

II. RELATED WORKS

Valuable activities have been done in the intrusion-detection area. Among these activities using a powerful modeling tool called hidden Markov model (HMM) is common. HMM is proper for modeling normal behaviors and based on that it can recognize noises and abnormal behaviors. This tool has been applied in other activities, like diagnosing movement and also speech. Corona and et al. [5] presented a new framework in which sent queries to web applications users are analyzed by HMM and has a special concern to noise existence in training data. Yung zhong Li [6] used a fuzzy approach for HMM in which the system is flexibly able to adapt with patterns change; thereafter, it has been improved in recognizing new

978-0-7695-4668-1/12 $26.00 © 2012 IEEE
DOI 10.1109/ISMS.2012.25

attacks. In order to recognize anomalies in system calling of system programs, Dau Xuan Hoang [10] defined a fuzzy schematic for intrusion-detection system in which fuzzy logic is used instead of using crisp and classic conditions. Ajith Abraham et al. [13] applied soft computing techniques, specifically fuzzy logic and neural network to construct a powerful and flexible intrusion-detection system. This system is based on multiple modeling with various measurements. And finally uses fuzzy logic for final decision making. C. Kruegel [9] introduced a multi model framework for diagnosing attacks on web applications that analyze received queries for both location and time properties. Among different models represented there HMM showed better results. Estevez Tapiador [11] presented a method based on supervision on received HTTP requests which through a Markov model, including a set of states and transition between them tries to diagnose attacks on web servers and decides about normality or being attack of a demand according to HTTP protocol characteristics.

III. IMPROVEMENT DETECTION ACCURACY

As it was mentioned, the main goal in this work is to decrease error rate. Error could consist of an attack which intrusion-detection system recognizes it as normal HTTP requests (false negative) or normal HTTP requests which the system labels them as an attack (false positive). Before discussing the way of increasing access diagnosis system' accuracy, two related subject with error and accuracy in diagnosing have to be noted. In intrusion-detection system, one HMM is usually trained for every HTTP request or every extracted feature from that and just that model is used for classifying HTTP requests. It causes that considered result to be influenced by that model's error. These errors might be generated because of improper training data set or using inappropriate training algorithms or improper configuration of HMM.

The issue is to apply a threshold value as the boundary between normal and abnormal HTTP request. Threshold value is obtained in training phase and by trial and error and doesn't act well for HTTP requests near classifying boundary and leads to increase the system error. As it is seen in figure 1, points of two normal and abnormal classes specified with dot and cross, respectively, are separated from each other by a line which this line is called decision boundary.

Those points which have located near the decision boundary in an area named grey area, are main candidates for errors, such that the smallest change in normal patterns results in increasing the false-positive rate. This situation is of high risk because the number of wrong alerts might increase by heavy traffic of HTTP requests. On the other hand, Fake attack existing near this boundary may be recognized as normal HTTP requests.

In following, two utilized techniques in this work are expressed, that increase accuracy and decrease detection errors in HTTP requests, which are multiple classifier systems and soft boundary.

A. Multiple classifier systems

Multiple classifier systems (MCS) have been vastly applied in pattern recognition problems in a way that in contrast to single classifier systems have shown much precision. Reasons can be observed in [3, 4]. A MCS could be a classifier ensemble, a combined classifier system, a multiple expert system and, etc. Although, a classifier ensemble is of higher accuracy in classifying than one classifier, it has to be mentioned that this accuracy increase leads to increase in system's complexity. Thereupon, it should trade off complexity against the accuracy increase in order to obtain an adequate accurate and few complex systems. Generally speaking, an MCS extracts decisions by a set of distinct classifiers and by combining them, reaches to a more accuracy classifier. At least, for one of the following reasons, an ensemble of classifiers is used instead of one classifier [12]: *Computational,* Statistical, Representational.

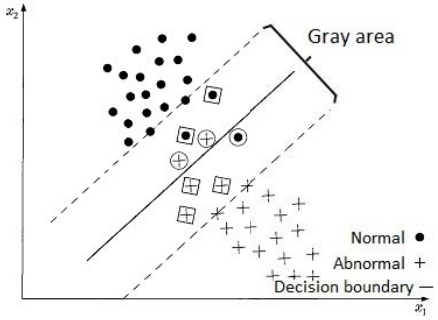

Figure1. classifying two normal and abnormal patterns. Points near the decision boundary locating in an area between two dash lines (gray) are main candidates of detection errors.

According to figure 2, here, multiple HMM is used and input x_i is sent to each of HMMs inside this ensemble. $H = \{HMM_j\}$ is composed of n HMM which every HMM_j produces output S_{ij}. Then, with combining outputs of every HMM by a classifier, final output S_i is produced.

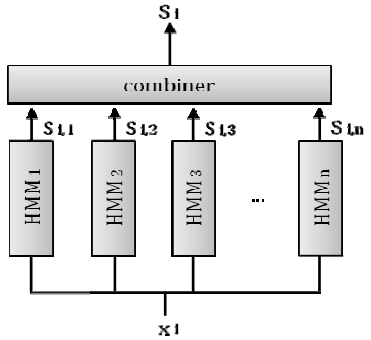

Figure2. MCS pattern. Input x_i is sent to every HMM, and their output is converted by a combiner to a considered output.

Two common combining strategies are fusion and selection which fusion approach applied in this paper will be explained.

Fusion: in this method, it was supposed that every ensemble's member has knowledge about the whole

feature's space. Outputs of every classifier in an ensemble are fused to produce an output. There are lots of fusion functions which each of them has some supporters and opponents. Here, the maximum, the minimum, the average and geometric average rules are shown:

$$\text{Maximum rule: } S_i^* = max\{S_{ij}\} \quad (1)$$

$$\text{Minimum rule: } S_i^* = min\{S_{ij}\} \quad (2)$$

$$\text{Average rule: } S_i^* = \frac{1}{N}\sum_{j=1}^{N} S_{ij} \quad (3)$$

$$\text{Geometric average rule: } S_i^* = \left[\prod_{j=1}^{N} s_{ij}\right]^{\frac{1}{N}} \quad (4)$$

These static rules have been so effective in pattern recognition even of their simplicity. Although combination rules with training methods give better results, for following reasons, static rules are used: speed of static rules in computations and building complexity of a "trained" combiner [20]. In this work, the maximum rule was used.

B. Soft boundary

One of the most important roles of anomaly detection is to determine between a normal and abnormal behavior of supervised objects. If this boundary is defined well, intrusion-detection system performs better in detecting attacks. But because of existing overlap between the normal and abnormal behaviors system wronged. This boundary is dividable into two hard and soft boundaries. Hard boundary is described as crisp classic conditions that an example of that is threshold value. In the contrary, soft boundary can be represented by fuzzy rules and sets. One of the techniques of soft computing in anomaly diagnosis is fuzzy logic that was discussed in [10]. Anomaly detection based on a soft boundary, or specifically on fuzzy sets and rules leads to better results for following reasons:

a. Since normality or abnormality is not an absolute concept, the definition of a hard boundary causes a sharp distinct between normality and abnormality. Hence, it is natural that it has to use fussy sets to define soft boundaries [15, 16]. In fuzzy logic, the degree of normality and also the degree of abnormality are determined by normalcy and abnormalcy, respectively.

b. Anomaly detection system based on fuzzy inference could combine received inputs from multiple sources which it leads to detection efficiency improvement [15].

IV. THE PROPOSED INTRUSION-DETECTION SYSTEM

A scheme of one the system's module is shown in figure 3. Generally, anomaly detection systems based machine learning algorithms are developed in two learning and testing (operational) phases and here is not out of this rule. In learning phase, system's models and modules are configured with respect to training data, which includes normal HTTP requests. In testing phase, the built system is evaluated with respect to test data set, which includes normal and abnormal HTTP requests. This system uses hidden Markov models to model a sequence of attributes and given value to them, which are received by web application. For ever web application in this system, a module is built, which consists of multiple HMM ensembles. Every HMM is trained for modeling each extracted feature from HTTP requests. Probable value resulted from HMMs is combined in an HMM ensemble and final considered output is obtained. Then, with fuzzifying the output of each HMM ensemble, inference and decision-making process becomes ready to be performed. In the inference process, by applying fuzzy sets and rules, which are generated according to web application, inference engine specifies normal and abnormal HTTP requests. In following, the proposed system is defined thoroughly.

A. Feature extraction and feature's security value

Here, it is focused on two important features of HTTP requests, which are:

1. *Attributes sequence*
2. *Given value to each attribute*

In this text, HMM is utilized as a modeling tool which its responsibility is to model extracted properties from every HTTP request. It is assumed that there exists a query, like *name=Mohammad&userID=21718*. Then this query has the attributes' sequence *<name,userID>* that has to be analyzed by hidden Markov model. As well, to assess the value of each of these attributes should be sent to hidden Markov model. For better understanding, a brief description is given for applied data model.

Data model: The proposed system' input is set of received uniform resource identifiers (URIs) by a web server which hosts considered web application, which they could be presented as $U=\{U_1, U_2, \ldots, U_m\}$. A URI, like U_i includes corresponding path to considered source ($path_i$), selected path to an informative component ($info_i$) and a query sequence (q_2). Among mentioned sections, query sequence is more affected by attacks. Figure 4 shows a HTTP request, which is in the log of a web server. According to figure 4, query sequence is after "?" which contains an ordered list of "attribute" and "value" pairs that are separated by "&". This list consists of the sent parameters to considered web applications. Consequently, it is possible to show the query as $q=\{(a_1, v_1),(a_2, v_2),\ldots,(a_n, v_n)\}$, in which $a_i \in A$ and A is the set of all attributed related to considered web application and v_i is a string value. Another important issue is that each query consists of a set of parameters with a special order, that is $S_{q_i} = \{a_j, \ldots, a_k\}$. On the other hand, every application, also, accepts specific queries. HTTP requests shown in figure 4 has a subset of attributes that are located with a specified order in the query: $\{a_1, a_2\}$. If you look up to web application, it is seen that every web application has a specified number of HTTP requests. Also, every HTTP request consists of attributes with a specific number and order, which each one of them has its own specific security value. For instance, in the above query, the attribute *userID* is of

more security value than the attribute n*ame*. However, it depends on web application.

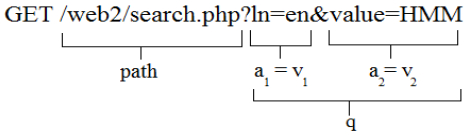

Figure4. HTTP request structure. Each HTTP request consists of a path, and a query and ever query contains parameters, which appear as a list of pairs of attribute and value in it.

Each of these attributes plays their role in the decision-making process according to their value. This value is determined according to considered web application. At the end, extracted attributes from HTTP requests have to be presented more simply to modeling module. For this purpose**,** it has to be discriminate between alphabetic-numeric characters and non alphabetic-numeric characters. Here, all alphabetic characters change to "*A*" and all numeric characters change to "*N,"* and others remain the same as before. As an example, previous query is converted into *name=AAAAAAAA&userID=NNNNN* which it is done in preprocessor module.

B. Modeling module

A module is built for every application on the web server. This module includes modeling and decision modules. In modeling modules, for every attribute that could be modeled, a HMM ensemble is used which every HMM in this ensemble is trained for modeling the corresponding feature in learning phase. Then the output of each of these HMMs is converted to the output that is the final result of HMM ensemble by a combination technique, like fusion. Moreover, it should be noted that multiple classifier techniques are complex and expensive and according to the value of every attribute, the complexity of its corresponding HMM has to be controlled. As an illustration, the number of HMMs which are in the ensemble related to the value of attribute u*serID* is more than the number of HMMs of the ensemble related to the value of attribute *name*.

Building HMM ensemble: Here, two problems of three problems are studied: one of them is learning problem (during learning phase), and the other is the evaluation problem. Baum-Welch algorithm [8] is used for training HMM. HMM efficiency depends on parameters, such as the number of states, initial state, symbols distribution matrix and states transition matrix. Since the optimum evaluation of HMM's parameters is an art instead of knowledge, an HMM ensemble is used to compensate this lack of knowledge. An ensemble of a number of equal states is applied for every HMM in a set. This number is equal to the average of training sequence's length (using round operation toward larger integer). The length of a sequence is determined by the number of different symbols. For example, in sequence {a,b,c,b,c}, three different symbols are seen that is a, b, c. So then, states correspond to elements in the analyzed sequence. Both state transition matrix and symbols distribution matrix are initialized randomly. The built system contains many HMMs and a priori knowledge is used to model matrixes structure that may need a lot of efforts and time. Combining HMM's output: there are determined solutions to combine HMM's outputs inside a set.

For a supposed input sequence *S*, the output of HMM i, m_i, which is expressed as:

$$P(s|m_i) = P(m_i|s)\, P(s) / p(m_i) \qquad (5)$$

For all models, the same priori probability is used:

$$P(m_i) = c\ ,\ \forall\ \ i \in [1,k] \qquad (6)$$

Where k is the total number of HMMs inside an ensemble. It is obviously seen when the maximum combination rule is used, that is:

$$\text{Output} = \max\ \{\ P(s|m_i)\ \}\ ,\ i \in [1,k] \qquad (7)$$

Where, the proper output with $i \in [1, k]$ is max $\{p(m_i|s)\}$ (p(s) is constant). Therefore, by the maximum rule, it could select a model that has the best possible description of analyzed sequences to calculate the sequence probability. In fact, this is the main reason of using HMM ensemble. In another word, to model much well, the whole set of training sequences, diversity of multiple HMM is utilized.

C. Decision module

Decision module's inputs are the outputs of applied HMM ensembles. Each output is the probability value that determines the normality of HTTP request with respect to its attributes. This module infers the output by applying fuzzy rules, that shows normality or abnormality of HTTP request. Figure 3 shows this module which has three steps of fuzzification, inference and defuzzification. In following, this module is described.

D. Building fuzzy sets

Fuzzy sets are built based on web application and unscientifically and with respect to observations. Each extracted attribute from HTTP requests has its own fuzzy sets according to its security value. For example, there is a HTTP request as display.aspx?name=mohamamd&userID=2345. Three properties are observed in this request, which consist of attributes sequence {name, userID}, attributes value of *userID* and *name* whose their security values decrease, respectively. Each of them has its own specific value and should be modeled. For instance, the value of attribute's sequence is more than the attribute's value of *userID*. There are four fuzzy sets for the feature of attribute's sequence that are very low, low, high and very high showing the probability resulted from HMM ensemble of attribute's sequence. For the feature of attribute's value of *userID*, there exist three sets low, medium and high, which show resulted probability from HMM ensemble for attribute's value of *userID*. Figure 5 is the graphically of membership functions of fuzzy sets corresponding mentioned properties. Hence, other fuzzy sets are built with respect to their security value.

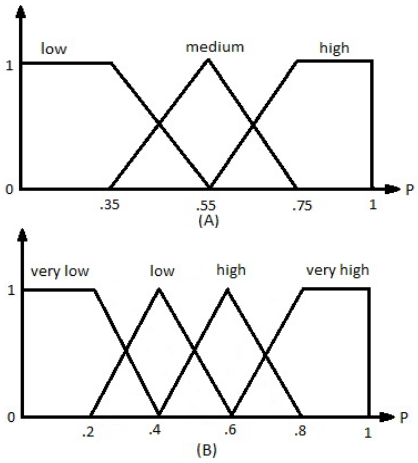

Figure5. Graph of membership functions of fuzzy sets. Figure A represents three fuzzy sets for the feature of attribute's value userID and figure B represents four fuzzy sets for the feature of attribute's order. Axis P states the output probability of an HMM ensemble.

E. Building fuzzy rules

Every web application has its own fuzzy rules that these rules are made based on extracted properties and their security value. For instance, in the example mentioned in last section, for an HTTP request, if a proper probability is obtained for *userID* and *name*, but the obtained probability for attributes sequence is improper, HTTP requests is abnormal. Alternatively, if obtained probability for attributes sequence and *userID* is enough and obtained probability for *name* is improper, HTTP requests is labeled as normal. So then, according to these assumptions, required rules are built. As an example:

- *IF attrSeq IS low THEN URI IS abnormal.*
- *IF attrSeq IS high AND userID IS medium AND name IS low THEN URI IS normal.*

F. Inference process

In fuzzification process, probable classic values of inputs are converted to fuzzy values, which are usually fuzzy set. Two important tasks done in this process are to convert probable classic values to fuzzy values in the form of fuzzy sets and to apply membership functions for computing truth degree of each adapted fuzzy set. In defuzzification process, fuzzy values are converted to real values that can be either of normal or abnormal values. There are many defuzzifier techniques, such as maximum defuzzifier, gravity center defuzzifier and centers average defuzzifier. In this paper, a maximum defuzzifier was used. In fuzzy inference step, created fuzzy rules locate in fuzzy rules base, are applied on pre-evaluated inputs to produce output. Therefore, after evaluating each introduction, it is combined by AND and OR and the minimum degree and the maximum degree are considered as the evaluation of antecedent part.

V. EXPERIMENTS AND EVALUATION OF THE PROPOSED SYSTEM

To evaluate the proposed system, twelve applications have been used, which four of them are for registration operation and logging into the system and the others present general services, like show, search and, etc. These applications are hosted in a server with Intel Core i7 microprocessor, 6 GB main memory, 6 MB cash memory, Windows server 2008 and IIS version 6. Required queries set is collected from the event register table of considered server. This set contains 80000 normal HTTP requests, which are collected from students of three universities in a one-month period. About forty percent of these requests are due to the first four applications, and the others are due to other applications. The proposed intrusion-detection system has a module for every application, and each module has multiple HMM ensembles according to extracted properties from HTTP requests to considered application, and each HMM ensemble has a specific number of HMM due to the security value of extracted feature. For example, for application display.aspx, a module named display is made. This module has three HMM ensembles, which are related to features of attributes sequence, value of attribute *name* and value of attribute *userID*. Corresponding HMM ensemble to value of attribute *userID* and *name* has eight and four HMM, Respectively. For simplicity, the number of HMMs' state in an ensemble is considered the same. Training data set is divided into five parts and every HMM is trained by one of those parts and is evaluated by other parts. To evaluate the proposed system, abnormal attack HTTP requests are needed. To collect the set of abnormal HTTP requests, from those attacks and published destructive codes in [19], 15 XSS attacks, 15 Code Injection attacks and 15 SQL Injection attacks were selected and along with 500, normal HTTP requests constitute an experimental set. Without noting that which algorithm has been used in an intrusion-detection system, it is possible to model it as a black box which receives an object from a specific source and determines whether an attack has occurred. That object's source could be a network or whatever else. The first important issue in assessing an intrusion-detection system is that how many of these objects were labeled correctly and how many incorrectly. Formally, attacks are usually recognized as positive class and normal objects as negative class. Consequently, four important definitions can be taken into considerations as criteria for evaluation of an intrusion-detection system:

1) *True positive:* It consists of an attack that was detected correctly by an intrusion-detection system. In fact, it is a sample of positive class, which was labeled correctly.

2) *False positive:* It consists of a normal object that was recognized incorrectly as an attack by the system.

3) *True negative:* a normal object which was labeled correctly and was classified in negative class.

4) *False negative:* an attack which has not been detected by the system and was classified into negative class, so it was incorrectly labeled.

Due to these definitions, in order to assess the

effectiveness of the proposed system it can use following evaluation expressions:

- False positive rate= $\frac{\text{number of negative objects which have been incorrectly labeled}}{\text{total number of objects in negative class}}$
- Detection rate=1- false negative rate
- False negative rate= $\frac{\text{number of positive objects which have been incorrectly labeled}}{\text{total number of objects in positive class}}$

Therefore, first, detection rate is evaluated and later, false-positive rate with having the maximum detection rate is evaluated. The results of this evaluation are in figure 6 and table 1. At last, various decision modules with fuzzy rules and sets are evaluated, which its results are in table 2.

To evaluate detection rate of two systems, one with an HMM and the other with an HMM ensemble are tested for each feature of the modeling module. According to figure 6, due to the maximum threshold, the system with HMM ensemble, with detecting 44 out of 45 attacks has better ROC than a system with one HMM with 39 attacks detection. But, between 500 normal HTTP requests, seven requests in the first system and 10 requests in the second system were incorrectly recognized as attacks. Hence, for the maximum threshold, although detection rate was so suitable, false-positive rates of both systems were inappropriate and cause systems' accuracy to be decreased.

However, along with increasing the number of HMMs inside an HMM ensemble, it could reach to a better curve which is obviously seen in [5]. Represented data in table 1 show the evaluation results of two systems, one with fuzzy decision module and the other with classic decision module (threshold) that HMM ensembles were used in both. Both systems were configured in a way that they have the maximum detection rates. It means that in the system with classic decision module, the most rigorous threshold was selected. The training set consists of 505 normal demands and five abnormal requests. The system with fuzzy decision module recognized correctly five attacks and just labeled incorrectly four normal requests as attacks. But another system with recognizing all attacks, diagnosed 16 normal requests incorrectly. So, false-positive rate of the system applying a fuzzy decision module is less than the system with classic decision module. Nevertheless, it should note that if fuzzy rules and sets are not well-defined and selected, they may have much weaker results. Finally, the proposed system with different fuzzy decision modules is evaluated so that various fuzzy sets and rules were used in each module. For this, intrusion-detection system has been limited to login application. With respect to table 2, fuzzy sets and rules of the decision module for login application became more complete and more precise, respectively.

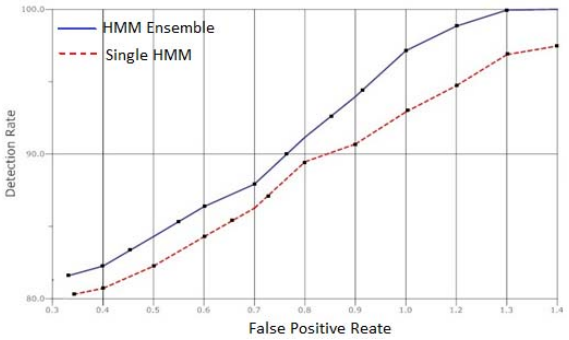

Figure6. RUC curves for two systems, one with one HMM and another with multiple HMMs in each ensemble. In both systems, decision module of threshold was applied. Black dots which are across from each other on both curves show different threshold, which by the continuous changes of threshold, attack detection rate of system increases and as a result, false-positive rate increases too. For the maximum threshold amount, the system with HMM ensemble by detecting 44 out of 45 attacks has better ROC than that one with 39 recognized attacks. But because of high false-positive rate in both systems, inappropriate diagnosis accuracy was resulted.

Table1. Comparing two systems with different decision modules. In the first column, the system with classic decision module (threshold) was evaluated for detection rate and false-positive rate and in the second column, the system with fuzzy decision module, 550 test HTTP requests were considered for evaluating two systems, which consist of 45 attack requests. As it is observed, the system with fuzzy decision module with the minimum false-positive rate and the maximum detection rate has better results.

IDS	With classic decision module	With fuzzy decision module
Number of detection attacks	45	45
Number of Normal requests, known attack	16	4
Detection rate	100%	100%
False-positive rate	3.16%	0.79%

Table2. Evaluation results of the proposed system with different fuzzy sets and rules in each decision module. All four systems were evaluated with 550 HTTP requests consisting of 45 attacks. The fist system in the first row with detecting 38 attacks correctly and 61 attacks incorrectly have the worst result and the system in the fourth row with detecting 44 attacks correctly and 10 attacks incorrectly have the best result. It is obvious that with increasing precision of fuzzy sets and rules, false-positive rate decreased considerably, and more abnormal demands were detected.

Login's decision module		Detection rate	False-positive rate
Fuzzy Rule	Fuzzy Set		
9	5	84.5%	12%
13	8	93.4%	6.5%
18	11	95.6%	3.4%
26	16	97.8%	1.9%

However, this evolution is impossible without the complexity increase in decision module, fuzzy sets and rules. The result of this complexity is decreasing system's speed and more usage of sources.

VI. CONCLUSION AND FUTURE WORKS

In this paper, an intrusion-detection system was introduced to detect malicious attacks on web applications that instead of using one HMM, multiple HMMs were used as an HMM ensemble to apply MCS technique for modeling extracted properties from sent queries to a web server, such that by fusing resulted outputs from an ensemble's HMMs, more accuracy to be obtained in modeling and evaluating the normality of those properties. On the other hand, this system has reached to the lowest false-positive rate by generating fuzzy sets and rules and applying fuzzy inference to decide about normality or being attack of one HTTP request instead of using a threshold. Thereafter, resulted output from each HMM ensemble having its own specific value are converted into fuzzy values. Then, inference engine determines normality or being attack of that HTTP request by receiving those fuzzy amounts and using fuzzy sets and rules generated manually. Applying these approaches leads to proper function for the proposed system in detecting new attacks. Furthermore, it behaves more flexibly with normal requests near the decision boundary and decreases false-positive rate to an acceptable level. During performing experiments in this work, the proposed with acceptable false-positive rate, 0.79% and detecting all available attacks on the training set has improved rather than an intrusion-detection system which has used the threshold for deciding (with false-positive rate 3.16%).

For future work, it can configure a decision module such as that fuzzy sets and rules to be affected by system function and to reach better results by auto or semi auto reorganizing them. As a matter of fact, create a learnable decision module.

REFERENCES

[1] Rfc2616, "hypertext transfer protocol," *http/1.1*, pp. 29-30.

[2] R. Auger et al, "Web security threat classification," *Web Application Security Consortium*, 2004.

[3] I. Corona, G. Giacinto, C.Mazzariello, F. Roli, and C. Sansone, "Information fusion for computer security: State of the art and open issues," *Information Fusion*, Vol. 10, Issue 4, pp. 274–284, 2009.

[4] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, and W. Lee, "Mcpad: A multiple classifier system for accurate payload-based anomaly detection," *Computer Networks, The International Journal of Computer and Telecommunications Networking*, Vol. 53, Issue 3, pp. 864–881, 2009.

[5] I. Corona, D. Ariu,and G. Giacinto, "HMM-Web: a framework for the detection of attacks against Web applications," *IEEE international conference on Communications*, Dresden, Germany, 2009.

[6] Yong zhong Li, Yang Ge, Xu Jing, and Zhao Bo, "A New Intrusion Detection Method Based on Fuzzy HMM," *ICIEA, IEEE Conference on*, 3rd, pp. 36-39, 2008.

[7] L.E. Baum, and J.A. Egon, "An inequality with applications to statistical estimation for probabilistic function of a markov process and to a model for ecology," *Bullettin American Metereology Society*, Vol. 73, No. 3, pp. 360-363, 1967.

[8] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, Vol. 77, Issue 2, pp. 257-286, 1989.

[9] C. Kruegel, G. Vigna, and W. Robertson, "A multi-model approach to the detection of web-based attacks," *Computer Networks*, Vol. 48, Issue 5, pp. 717–738, 2005.

[10] Dau Xuan Hoang, and Minh Ngoc Nguyen, "A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference," *Journal of Network and Computer Applications*, Vol. 32, Issue 6, November 2009.

[11] Estevez Tapiador, Garcia Teodoro, and Diaz Verdejo, "Detection of Web-based Attacks through Markovian Protocol Parsing", *10th IEEE Symposium on Computers and Communications*, pp. 457-462, 2005.

[12] R.O. Duda, P.E. Hart, and D.G. Stork, "Pattern Classification," *Wiley*, pp. 10-40, 2000.

[13] Ajith Abraham, Ravi Jain, "Soft Computing Models for Network Intrusion Detection Systems", *Classification and Clustering for Knowledge Discovery Studies in Computational Intelligence*, Vol. 4, pp. 191-207, 2005.

[14] Ghmm: General hidden markov model library, http://ghmm.org/.

[15] J.E. Dickerson, J. Juslin, O. Koukousoula, and J.A. Dickerson, "Fuzzy Intrusion Detection," *IFSA World Congress and 20th NAFIPS International Conference on*, Vol. 3, pp. 1506-1510, Vancouver, Canada, 2001.

[16] J. Gomez, F. Gonzalez, and D. Dasgupta, "An Immuno-Fuzzy Approach to Anomaly Detection," *Fuzzy Systems*, 12th IEEE International Conference on, Vol. 2, pp. 1219-1224, 2003.

[17] L.A. Zadeh, "Fuzzy sets," *in the Information and Control Journal*, Vol. 8, page 338, 1965.

[18] E. Cox, "Fuzzy fundamentals," *Spectrum*, IEEE, Vol. 29, No. 10, page 58, 1992.[19] milw0rm, Web application and HTTP attacks published database, www.milw0rm.com.

[20] R.P.W. Duin, "The combining classifier: to train or not to train? In Pattern Recognition," *16th International Conference on*, Vol 2, pp. 765–770, 2002.

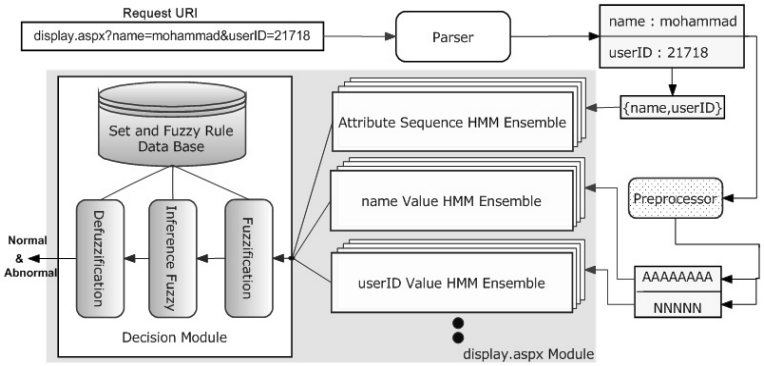

Figure3. A schematic of the proposed system