Using Fuzzy Decision Trees for Authentication via Keystroke Timings

M. R. Goodarzi
Department of Computer Engineering
Iran University of Science and Technology (IUST)
mrgoodarzi@comp.iust.ac.ir

M. R. Kangavari
Department of Computer Engineering
Iran University of Science and Technology (IUST)
kangavari@iust.ac.ir

Abstract – **Decision trees are one the most popular methodologies for acquiring knowledge from raw data. The resulting knowledge, in the form of a symbolic decision tree along with a simple inference mechanism, has been praised for comprehensibility. Fuzzy decision trees merge decision tree learning algorithms and fuzzy representation with the aim to better deal with inconsistent, noisy, and inexact data. In this paper a new method of authenticating users via keystroke timings is presented that uses fuzzy decision tree learning method.**

I. INTRODUCTION

The problem of authenticating users in a computer system has been a topic of study since the inception of multi-user computing environments. There are a number of methods for authenticating users, the most common and the simplest one is using username/password mechanism. However, this method of authentication relies on the secrecy of the password and, in some cases, secrecy of the username. Because username/password pairs are easily transferable from one user to another, whether transferred inadvertently or not, this type of authentication suffers form a notable weakness and therefore it's not sufficient for high secure environments.

As a complement (or replacement) of the traditional password-based systems, there have been developed many systems that are based on the use of the users' specific physiological and biological data (e.g.[1]-[4]). Traditionally, using picture IDs and handwriting signatures as a means of authentication/identification, had been popular in the finance and banking systems for a long time. Such systems have shown to be successful because they are based on data that is largely unique to individuals.

The machine-measurable features of physiological and biological data are collectively known as biometrics [5]. Some common biometrics include voice, iris patterns and fingerprints.

Most biometric technologies require special hardware to convert along measurements of signatures, voices, or patterns of fingerprints and palm prints, to digital measurement, which computers can read. This results in security systems that are usually expensive to deploy and use.

However, there is another type of biometric that could be used as a «cheap» biometric in the authentication systems. Proposed as early as 1980 [6], this kind of biometric relies on how the user types in the keys, not only what he/she types [7]. It offers a big advantage that requires no special hardware for data acquisition other than the keyboard. Furthermore, the process is practically invisible to the user, since the user is merely asked to type his/her password while the keystrokes are recorded. In addition, the technique is highly flexible, as it accommodates to the password changes.

Gaines *et. al.* [6] were the first to investigate the possibility of using keystroke timings for authentication. A test of statistical independence of seven users profiles was carried out using the T-Test under the hypothesis that for a particular user the means of the digraph times at two sessions were the same, but the variances were different. Similar experiments were conducted by Legget *et. al.* ([8]-[9]) with seventeen programmers but for the continuous approach to user verification. The authors reported an identity verifier system with a false alarm rate (i.e. the rate a legitimate user is rejected) of about 5.5 percent and imposter pass rate (i.e. the rate an imposter could be verified) of approximately 5.0 percent. Later, Joyce and Gupta [10] were able to show an imposter pass rate of 0.25 percent (2 out of 810 unauthorized attempts gained access) and a false alarm rate of 16.36 percent (27 out of every 165 valid attempts were denied).

While all of the above-mentioned works are based on using the statistical methods, there are a few works in the literature that use pattern recognition techniques like Neural Networks ([11]-[13]), Markov models ([14]), and K-Nearest Neighbours ([15]) to solve this classification problem. In this paper we will show that decision tree classifiers can be used to solve the problem as well.

Decision tree algorithms provide one the most popular methodologies for acquiring knowledge from raw data. The extracted knowledge, which is represented by a decision tree structure, is praised for high comprehensibility. Such knowledge could then be employed in tasks such as decision-making, decision justification, classification and so on. There are many decision tree algorithms [22]. ID3 [16], and its successor C4.5 [17], along with CART [18], are the most popular methods that have shown successful in many domains.

However, in learning symbolic decision trees some difficulties may arise if data is noisy, erroneous, incomplete, and inconsistent or if it comes with some subjectivity / ambiguity [19]. These characteristics are common to many real-world applications. Symbolic decision trees usually have a poor performance in domains where there are a lot of noise, error, inconsistency and uncertainty.

In recent years, using fuzzy representation in dealing with such situations has become very popular. Fuzzy representation, based on fuzzy sets and used in approximate reasoning, is especially applicable to bridging the conceptual gap between subjective/ambiguous features and quantitative data [20]. Its also well suited for dealing with inexact and noisy data. Fuzzy rules, based on fuzzy sets, utilize those qualities of fuzzy representation in a comprehensible structure of rule bases. This makes fuzzy systems ideal to be used in many practical applications.

Fuzzy decision trees (FDTs) are a special case of fuzzy systems that result from merging symbolic decision trees with fuzzy representation. They have been proposed by a few researchers with the aim to preserve the advantages of both decision trees and fuzzy representation: the comprehensibility, popularity and ease-of-use of the former and the gradual processing and uncertainty handling capabilities of the latter [21]. Fuzzy decision trees have also been used in many applications.

This paper presents a new method of authentication via keystroke timings that uses fuzzy decision trees methodology. As we will see, fuzzy decision trees provide us with a highly accurate method that could reliably be used in computer security systems.

The rest of this paper is organized as follows. After a review of symbolic decision trees in section II, and fuzzy decision trees in section III, description of the experiments and the obtained results is presented in section IV. Finally, Section V concludes the results and makes some suggestions for future work.

II. DECISION TREES

Decision tree induction has been studied in detail both in the area of pattern recognition and in the area of machine learning. Various heuristic methods have been proposed for designing a decision tree [22]. Among them, Quinlan s ID3 [16] is the most widely known. It was originally designed for symbolic data. The acquired knowledge is expressed with a highly comprehensible symbolic decision tree, which paired with a simple inference mechanism assigns symbolic decisions to new data. Because of the natural interpretation of the knowledge, symbolic decision trees can be easily translated to a set of rules suitable for use in rule-based systems [17].

The ID3 tree-building procedure in fact creates a hard partition of the description space and represents the partition as a tree. It tries to make partition blocks as big as possible while each of the blocks contains samples of only one class. An example of a resulting decision tree, built from examples of two classes, illustrated as black and white, is presented in Fig. 1.

In the inference stage, features of a new example are matched against the conditions of the tree. This of course corresponds to deciding on the partition block that the new example falls into. The classification of the examples of that leaf whose conditions are satisfied by the data is returned as the algorithm s decision. For example, a new sample with the following features: [*Color=Yellow*][*Stuff=Wood*] matches the conditions leading to *L4* in Fig. 1. Thus, it would be assigned the black classification here.

ID3 has shown to be successful in many symbolic domains. However there are a lot of real-world applications in that some or all of the attributes are continues-valued or multi-valued and not symbolic. This may require an a priori partitioning in the domains of such attributes. On the other hand, there are some modifications of the ID3 algorithm (e.g. C4.5 [17]) that does not require any prior partitioning. The conditions in the tree are based on thresholds (for continuous domains), which are dynamically computed. Furthermore the conditions on a path can use a given attribute a number of times (with different thresholds), and the thresholds used on different paths are very likely to differ. Although this usually increases the tree quality, but at the same time it may reduce its comprehensibility.

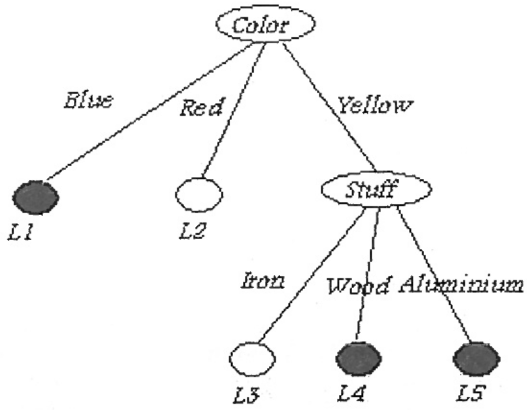

Fig. 1. An example of a symbolic decision tree

In addition to the above problem, real-world data usually contains some noisy, erroneous or inconsistent samples. Also some features or their values may be missing from data descriptions, and this may happen both in training and decision-making stages. Such problems lead to building trees that are incomplete (missing nodes) or inconsistent (having leaf nodes containing training examples of non-unique classes or having distinct leaf nodes intersecting in some training examples). An insufficient set of training examples may be another cause of creating incomplete trees. Inconsistent and incomplete trees may lead to difficulties in the inference stage, namely no-match and multiple-match problems. The no-match problem occurs whenever none of the leaf nodes of the tree is satisfied. The multiple-match problem appears whenever a leaf node is satisfied that contains examples of non-unique classes or whenever multiple leaf nodes are satisfied, while they have non-unique classifications.

Some of these potential problems in decision trees have been extensively addressed in the literature. For example, Quinlan has suggested that in tree-building, when an attribute has its information contents computed in order to determine its utility for splitting a node, each example whose needed feature is missing be partially matched, to the same normalized degree, to all conditions of the attribute. However, the overall utility of the attribute should be reduced according to the proportion of its missing features [23]. During inference, he suggested that assuming similar partial truth-ness of conditions leading to all available children gives the best performance [23]. This, however, leads to inconsistencies (multiple, possibly conflicting leaves could be satisfied), and a probabilistic combination must be used [17]. Quinlan has also investigated the behaviour of such a probabilistic approach on noisy features [17,24]. Such modifications increased the applicability of decision tree learning methods in various domains while at the same time improved the qualities of built trees in terms of accuracy and comprehensibility. In recent years, some researchers proposed to use fuzzy decision trees as a better learning approach to overcome the real-world applications problems.

III. FUZZY DECISION TREES

Most knowledge associated with humans thinking and perception has imprecision and uncertainty. In addition to the experience of domain experts, learning from examples with fuzzy representation is considered as an essential way of acquiring such knowledge.

Fuzzy decision trees are hybrid machine-learning approaches that result from incorporating fuzzy representation into traditional decision tree learning algorithms. There are a number of methodologies for building fuzzy decision trees (e.g. [21], [25]-[27]). The method described here is an extension of ID3 algorithm for learning symbolic decision trees [26]. Here examples are considered to have a fuzzy representation. A fuzzy decision tree (FDT) differs form a crisp decision tree (CDT) in the following several aspects [28]:

1) A FDT is a fuzzy partition of X while a CDT is a crisp partition of X, where X is the universe of discourse of all training examples.

2) Each node of the FDT is a fuzzy set defined on X while each node of the CDT is a crisp set of X.

3) The intersection of nodes located on the same layer is nonempty in FDT but is empty in CDT.

4) In the fuzzy case, if N is non-leaf node and $\{N_j\}$ is the set of all son-nodes of N, then $\cup_j N_j \subset N$. In the crisp case, the equality $\cup_j N_j = N$ holds well.

5) Each attribute-value is regarded as a fuzzy set in fuzzy case but as a crisp set in crisp case.

6) Each path from the root to a leaf can be converted to a fuzzy rule with some degree of truth in fuzzy case, but a crisp production rule in crisp case.

7) An example remaining to be classified matches only one path in the CDT, but may match several paths in the FDT.

The Fuzzy ID3 algorithm [26] for building a fuzzy decision tree follows that of ID3, except that information utility of individual attributes is evaluated using fuzzy sets, memberships, and reasoning methods. More specifically, the difference lies in the way the probabilities p_k are estimated (where p_k denotes the probability that a sample in node N belongs to class C_k). These probabilities are estimated from example counts in individual nodes. In fuzzy decision trees, an example s membership in a given node is a real number in [0,1], indicating the example s satisfaction of the fuzzy restrictions leading to that node. Denote $\mu_N(e)$ as the accumulated membership for example e at node N. Clearly, $\mu_{Root}(e)=1$, and $\mu_{N_j}(e)=t(\mu_N(e),\mu_{A_j}(e))$, where N_j is the jth child of N, A_j is the fuzzy term associated with the fuzzy restriction leading to N_j and $t(\cdot,\cdot)$ is a *T*-norm operator (usually *product* or *min*). Then, given that $f(\cdot,\cdot)$ denotes an appropriate operator for the fuzzy implication of a rule (which is often a *T*-norm), at node N the probabilities p_k can be estimated as:

$$p_k = \frac{\sum_{e\in E} f(\mu_N(e),\mu_{C_k}(e))}{\sum_{k'=1}^{|C|}\sum_{e\in E} f(\mu_N(e),\mu_{C_{k'}}(e))}, \tag{1}$$

and a tree can be constructed using the same algorithm as that of ID3.

There have been proposed many inference procedures for the fuzzy decision trees ([19], [21] and [26]). Here we adopt the one proposed by [26]. For this purpose, after the tree is built, we must assign an appropriate label to the tree leaves. In FDTs there are likely to be conflicts in individual leaves. So, one may decide to label a leaf node with the common classification of its examples. A better approach is to account for all the examples and label the leaf as follows:

$$CL(N)=\{(C_k,p_k)|k=1,2,\mathrm{K},|C|\}, \tag{2}$$

Now for a new sample to be classified it's required to compute the membership degree of the new sample in each of the leaves. In other words, it is now necessary to find its membership degree in each of the partition blocks of the now fuzzy partition. This membership degree is computed using the same algorithm that determines the membership degree of a training sample in a node of the tree. These memberships determine which of the leaves are satisfied by the sample. In FDTs the number of satisfied leaves dramatically increases. An aggregation operator must be used to appropriately combine the satisfaction degrees of all the leaves. Thus the membership degree of the new sample e in a class C_k can be computed using the following relation:

$$\hat{\mu}_{C_k}(e)=\sum_{N\in L} g(\kappa_N(c_k),\mu_N(e)), \tag{3}$$

In the above equation, L is the set of all the leaves, $g(\cdot,\cdot)$ is a fuzzy implication operator (usually product) and $\hat{\mu}_{C_k}(e)$ is the inferred membership degree of the sample e at class C_k.

Fuzzy decision trees have shown to be more accurate than symbolic decision trees in many domains. In the next section we describe an application of fuzzy decision trees to user authentication via keystroke timings

IV. EXPERIMENTS

A low-level program was used to record, with ms. precision, the stroke happening times. Each user had to repeatedly type the sentence: "This is a test.↵". During type of a sentence the low-level program was recording the press/release times of the typed keys. This way a number of time sequences were obtained that we refer to as "Exact" times. These exact times were used to extract two other time sequences: "Duration" times and "Interval" times. Duration times (or latencies) are the time intervals of holding the keys down, while Interval times are the time intervals passed between releasing a key and pressing its next key.

Participation in the experiment was open. Eight users chose to participate in the experiment. Half of the users were considered to be 'Right" users (or valid users) and the others were considered as 'Bad" users (in valid users or imposters). Table I shows distribution of samples in each of the two groups.

After collecting data, the experiment was conducted using 10-fold cross validation method. More specifically, 2/3 of the data samples were randomly selected as training set and the remaining 1/3 were used as test set. Using the training set, three types of learners were made: a fuzzy decision tree following the method mentioned in previous section, a symbolic decision tree using C4.5 program release 8 [29], and a K-NN classifier. To obtain a fairly unbiased estimate of the learners' predictive accuracy, each of the learners were separately tested on the test set. This process was repeated 10 times and the average of the recorded accuracy estimates was computed. Table II shows the results.

From table II, one can see that for each of the three methods the results of tests with Interval times shows a better accuracy than Duration times. Furthermore the best result is obtained for the Interval data with the fuzzy decision tree learning method.

Now a second experiment is performed on the Interval dataset in order to find which of the attributes have the most profound effect on discriminating the users. In this experiment each time a fuzzy decision tree is built those attributes that are cited near the root node (i.e.. attributes cited in the first and second levels of the tree) are recorded along with their presence frequency. The experiment is repeated 10 times and the obtained statistics is recorded in a table like table III.

The entries in table III are interpreted as follows. In all of the constructed trees the *A12* attribute is always the test attribute at the root node. In level 2 the attributes *A2* and *A3* are selected as the test attribute each 35% of the times and the remaining 30% is allocated to *A6*. Other attributes don't have any contribution in these two levels. Since in decision trees the most discriminator attributes are selected in the highest levels of the tree, from table III, it can be inferred that the most discriminator feature in the time intervals of the sentence: "This is a test. ↵" is the time interval between releasing the <*e*> key and pressing the <*s*> key, which for the sake of simplicity we represent it with notation $e \rightarrow s$. After this feature, the most three important features are respectively $h \rightarrow i$, the first $i \rightarrow s$ and the second $i \rightarrow s$.

TABLE I

DISTRIBUTION OF SAMPLES OF THE EIGHT USERS IN EACH OF THE TWO GROUPS (R1 MEANS RIGHT USER #1, B#1 MEANS BAD USER #1 AND SO ON)

Right Users	(R1) 462	(R2) 140	(R3) 114	(R4) 130
Bad Users	(B1) 151	(B2) 163	(B3) 140	(B4) 209

TABLE II

PREDICTIVE ACCURACY OF DIFFERENT METHODS

Dataset	FDT	C4.5	K-NN
Duration	80.8	80.0	80.9
Interval	94.1	88.8	84.6

TABLE III

THE MOST DISCRIMINATOR ATTRIBUTES

Tree Level	A2	*A3*	*A6*	A12
Level1 (Root)	-	-	-	100%
Level2	35%	35%	30%	-

V. CONCLUSIONS

The results of using fuzzy decision trees on the Interval dataset are promising. Here we obtained a 5.9% classification error rate on the test datasets. Our studies showed that distribution of errors in the two classes 'Bad" and 'Right" were nearly the same, thus we achieved an imposter pass rate of about 2.95% that is equal to the false alarm rate, which is about 2.95% too. These rates make this method highly reliable in order to be used for user authentication.

Crisp decision trees resulted a lower classification accuracy than fuzzy decision trees, probably because there were some inconsistencies in the data. That is there existed some records belonging to different users that were very similar to each other.

In addition to achieving high classification accuracy, we obtained some insights about features that have most discrimination power among others. This knowledge is highly valuable and will help us to find the most effective stroke timings that could uniquely identify each person.

Although we obtained good classification results in our experiments, but similar experiments must be conducted on other datasets comprising samples of a larger population of people, to find whether this method is applicable for systems having more users.

Authentication systems usually try to minimize the imposter pass rate. There are decision tree methodologies in that misclassification of samples of each class could be associated with a specific cost. A good idea for achieving lower imposter pass rate, at the cost of having a higher false alarm rate, is to use such cost-sensitive decision tree learning algorithms.

It must be noted that we only studied the proposed method on the two datasets of Duration and Interval. It's also a good idea to test weather or not, a combination of this two could result better performance.

VI. REFERENCES

[1] G.R. Doddington, 'Speaker Recognition – Iidentifying People by their Voices," in *Proceedings of the 1985 IEEE*, vol. 73, no. 11 , pp.1651-1664.

[2] D.K. Isenor and S.G. Zaky, 'Fingerprint identification using graph matching," *Pattern Recognition*, vol. 19, no. 2, 1986, pp.113-222.

[3] J. Zhang, Y. Yan, and M. Lades, 'Face recognition: eigenface, elastic matching and neural nets," in *Proceedings of the 1997 IEEE*, vol. 85, no. 9, pp.1423-1435.

[4] K,L. Kroeker, 'Exploring Visual Biometrics," *IEEE Computer Graphics and Applications*, July/August 2002, vol. 22, no. 4, pp. 16-21.

[5] S. Nanavati, M. Thieme and R. Nanavati, *Biometrics: Identity Verification in a Networked World*, Wiley Computer, NY: 2002.

[6] R. Gaines, W. Lisowski, S. Press, and N. Shapiro, Authen tication by Keystroke Timing: Some Preliminary Results, *Rand Report R-256-NSF*, Rand Corporation, Santa Monica, CA: 1980

[7] F. Monrose and A. Rubin, Keystroke Dynamics as a Biometric for Authentication, *Future Generation Computer Systems,* vol. 16, no. 4, Feb. 2000, pp. 351-359.

[8] J. Legget and G. Williams, Verifying idenity via keystroke characteristics, *International Journal of Man-Machine Studies*, vol. 28, 1988, pp.67-76.

[9] J. Legget, G. Williams, and D. Umphress, Verfication of user identity via keystroke characteristics, *Human Factors in Management Information Systems*, 1988.

[10] R. Joyce and G. Gupta, Identity Authorization Based on Keystroke Latencies, *Communications of the ACM*, vol. 33, no. 2, Feb.1990, pp. 168-176.

[11] B. Hussien, R. McLaren, and S. Bleha. An application of fuzzy algorithms in a computer access security system, *Pattern Recognition Letters*, vol. 9, no. 1, 1989, pp.39-43.

[12] M. Brown and S.J. Rogers. User Identification via Keystroke Characteristics of Typed Names using Neural Networks, *International Journal of Man-Machine Studies*, vol. 39, no. 6, 1993, pp. 999-1014.

[13] T. J. Alexandre, Biometrics on smartcards: an approach to keyboard behavioral signature, in *1996 Second Smart Card Research ffl Advanced Applications Conference*.

[14] D. Song, P. Venable, and A. Perrig, User Recognition by Keystroke Latency Pattern Analysis, available at http://paris.cs.berkely.edu/perrig/projects/keystroke/file.ps

[15] A. Peacock, Learning User Keystroke Latency Patterns , available at http://pel.cs.byu.edu/~alen/personal/CourseWork.cs572/KeystrokePaper/

[16] J.R. Quinla, Induction on Decision Trees, *Machine Learning*, vol.1, 1986, pp. 81-106.

[17] J.R.Quinlan, *C4.5: Programms for Machine Learning*, Morgan Kaufman, 1993.

[18] L. Breiman, J.H. Friedman, R.A. Oslan, and C.J.Stone, *Classification and Regression Trees*, Belmont, CA: Wadsworth, 1984.

[19] C.Z. Janikow, Exemplar Learning in Fuzzy Decision Trees, in Proceedings of the 1996 IEEE International Conference on Fuzzy Systems, pp. 1500-1505.

[20] L.A. Zadeh, Fuzzy Logic and Approximate Reasoning, *Synthese30*, 1975, pp. 407-428.

[21] C.Z. Janikow, Fuzzy Decision Trees: Issues and Methods, *IEEE Trans. Systems, Man, and Cybernetics*, vol. 28, no. 1, Feb. 1998., pp. 1-14.

[22] S.R. Safavian, and D. Landgrebe, A Survey of Decision Tree Classifier Methodologies, *IEEE Trans. Systems, Man, and Cybernetics*, vol. 21, no. 3, 1991, pp. 660-674.

[23] J.R. Quinlan, Unknown A ttribute-Values in Induction, in *Proceedings of the sixth International Workshop on Machine Learning*, 1989, pp. 164-168.

[24] J.R. Quinlan, The effect of Noise on Concept Learning, *Machine Learning II*, R. Michalski, J. Carbonell & T. Mitchel (eds), vol. 2, 1986, pp. 149-166.

[25] J, Jang, Structure Determination in Fuzzy Modeling: A Fuzzy CART approach, in *Proceedings of the 1994 IEEE Conference on Fuzzy Systems*, pp. 480-485.

[26] M. Umano, H. Okamoto, I. Hatono, H. Tamuro, F. Kawachi, S. Umedsu, and J.Kinoehita, Fuzzy Decision Trees by Fuzzy ID3 Algorithm and Its Application to Diagnosis Systems, in *Proceedings of The 1994 IEEE Conference on Fuzzy Systems*, pp. 2113-2118.

[27] C. Olaru and L. Wehenkel, A Complete Fuzzy Decision Tree Technique, *Fuzzy Sets and Systems*, vol. 138, no. 2, Sep. 2003, pp. 221-254.

[28] E.C.C. Tsang, X.Z. Wang, and D.S. Yeung, Improving Learning Accuracy of Fuzzy Decision Trees by Hybrid Neural Networks, IEEE Trans. Fuzzy Systems, vol. 8, no. 5,Oct. 2000, pp. 601-614.

[29] C4.5 Release 8 , available at: http://www.cse.unsw.edu.au/~quinlan/