Automatic Detection of Student Off-Task Behavior while Using an Intelligent Tutor for Algebra

Allan Edgar C. BATE
Department of Information Systems and Computer Science
Ateneo de Manila
Quezon City, Philippines
allbate@gmail.com

Ma. Mercedes T. RODRIGO
Department of Information Systems and Computer Science
Ateneo de Manila
Quezon City, Philippines
mrodrigo@ateneo.edu

***Abstract*— As more and more modern classrooms use Intelligent Tutoring Systems, it becomes imperative for our educators to determine whether these systems are being used properly, or whether students engage in off-task behavior. Off-task behavior is defined as disengagement from a learning experience. It can range from resting one's eyes to talking to one's seatmate. It can also take the form of "gaming the system" defined as attempting to advance through the curriculum by abusing regularities in the system. Gaming is operationalized as systematic guessing or trial and error. These behaviors constitute time away from the learning task and are therefore considered detrimental to learning. In this study, we recorded student interactions with Aplusix, an intelligent tutor for algebra. We then asked two experts to label excerpts or clips of these interactions. Finally, we used machine learning techniques to create detectors of off-task behavior.**

Keywords- Affective Computing, Intelligent Tutoring Systems, Machine-learning, Aplusix, Off-task behavior

I. INTRODUCTION

Intelligent tutoring systems (ITSs) are a subtype of computer-based learning system that make use of artificial intelligence to increase teaching effectiveness. ITSs are designed to provide students with individualized explanations, exercises, and remediation to help them learn the curriculum within the ITSs' domain [5]. Past uses of ITSs found that ITSs do increase student achievement by up to 100% [cf. 4].

Support for this project was provided by the Science Education Institute - Department of Science and Technology (SEI-DOST) through the Engineering Research and Development for Technology (ERDT) program.

Despite their known benefits, ITSs are still prone to inappropriate usage. One form of inappropriate use is off-task behavior.

A. *Statement of the Problem*

Off-task behavior is a defined as disengagement from a learning experience [7]. It is associated with poor learning. Baker et al [1] found that the students who engaged in off-task behavior during the use of an intelligent tutor learned only two-thirds of the subject matter, as compared to students who used the tutor properly.

B. *Goal*

In order to prevent the loss of learning opportunities for the student while using ITSs, we attempt to create a model that will detect off-task behavior during the students' use of the ITS. We achieve this through the labeling and analysis of data logs that contain student interactions during use of the ITS.

C. *Research Questions*

In this study, we hope to answer the following questions:

1. What information do we need to have a significantly valid low-fidelity text replay of the use of the ITS?

2. What are the different patterns of behavior that display off-task behavior of the student?

D. *Significance*

In traditional classrooms, teachers are able to identify when students start to lose interest and know when to intervene in order to correct them. For ITSs, we believe that it is important to be able to detect when students begin to lose interest and start engaging in off-task behavior. Automatic detection

enables ITS designers to devise interventions when faced with a student who is off-task. As Baker et al mentioned in [1], ITSs' responses to off-task behavior is an interesting area research as these responses may affect learning. If we can detect and prevent student off-task behavior, we may be able to increase the learning gained and the efficiency and effectiveness of ITSs.

II. THEORETICAL FRAMEWORK

We embarked on this study on the assumption that "cognitive processes can…be inferred from studying and comparing types of overt behavior. Log file analysis can be used when the purpose is to infer the cognitive processes of persons who interact with a computer program." [3]

A. *Log File Analysis*

Log file analysis is the systematic approach to examining and interpreting the content of behavioral data [3]. Log file analysis approaches include:

Transition analysis refers to the analysis the changes in behavior.

Frequency analysis refers to the tallying of frequencies of actions and computing for different statistics such as averages, and standard deviations.

Learning-indicator approach, similar to frequency approach, consists of clustering actions that have close-to-similar frequencies and determine groups in a global coverage.

Sequence analysis pays more attention to the belief that actions are the results of the actions before it and the reasons for the actions after it.

B. *Low-fidelity text replays*

Low-fidelity text replays [2] are clips of student actions. They are said to be low-fidelity because they make use of text alone as opposed to higher-fidelity media video or biometrics. Low-fidelity text replays have been shown to carry enough data to enable experts to make accurate inferences about student off-task behavior [2]. For this study, we show text replays to experts and ask these experts to label these clips as indicating off-task or on-task behavior.

We begin our study of student actions with a sequence analysis approach. We divide interaction logs into excerpts or clips that we hope illustrate whether a student is off-task or not. We ask experts to label these clips. Finally, we perform a frequency analysis to determine whether clips in one category differ from clips in another.

III. METHODOLOGY

Aplusix[2] is an ITS for Algebra and it presents students with exercises on varying math topics and levels of difficulties. Students solve these problems in a stepwise fashion, as they would on paper (See Figure 1). Aplusix records any interaction the student makes, from keystrokes to mouse clicks.

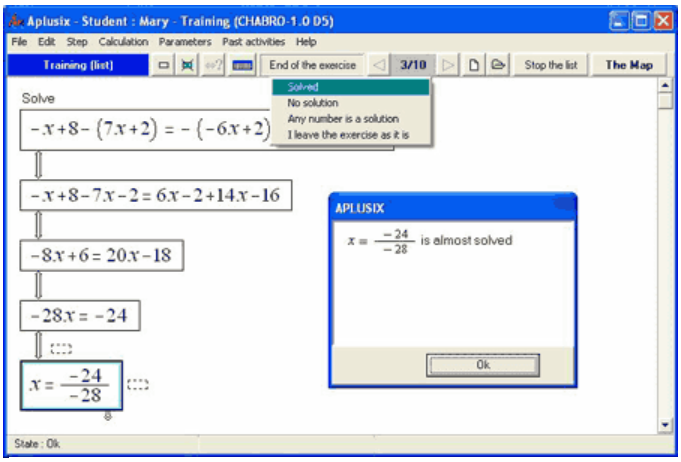

Figure 1. A screenshot of Aplusix II of solving a problem in a series of equations in step boxes.

A. *Data Distillation from Aplusix*

We obtained log files generated by Aplusix during the exercises of a previous research by Rodrigo et al. [6]. Among the data provided, we selected six parameters:

- **Turn** - a counting number assigned for each action that the student performed.
- **Time** - the number of seconds between the previous action and the current action.
- **Action** - the keystroke or mouse click that the student enacted.
- **Step** - the step box (see Figure 1) of the equation where the action was performed.
- **Expression** - the final state of the equation after making the action.

[2] http://www.chartwellyorke.com/aplusix/index.html

- **Status** – an indicator of whether the equation was solved or not.

We divided the logs into 20-second clips. From a population of 11,220 clips, we randomly selected a sample of 391 clips for labeling. We then showed each clip to our experts (Figure 2) for them to classify.

B. *Labeling*

We asked Dr. Cornelia Soto of the Ateneo de Manila University's Education department and Mrs. Ria Arespacochaga of the Ateneo High School Math Department to serve as our experts. Both have extensive knowledge of math education and off-task behavior in classrooms. They identified which behavioral patterns will tell us if the student is being off-task.

C. *Machine Learning Using WEKA*

Our classified clips were summarized into vectors containing the labels supplied by our experts and the features concerning each clip. Feature reduction was performed to further optimize the machine learning. The J48 algorithm supported by WEKA, was used and gave us an output of a C4.5 decision tree, and was validated using the 10-fold cross validation.

Move	Step	Events
		The student performs **0 action(s)** prior to this clip.
2	1	The student moves to step 1 and performs: **Duplicate current step**. 9x-(9-(-2x+8)) **The step has equivalence** The problem is **not solved**
3		The student moves the cursor with: **Place cursor** after **3.70 second(s)**.
4		The student begins to delete with: **BackSpace** after **1.30 second(s)** for the next **3 turns** in **1.1 second(s)**. 9x-(92x+8) The step has a **non-equivalence** The problem is **not solved** and has **non-equivalences**
7		The student types: **+** after **1.60 second(s)**. 9x-(9+2x+8) The step has a **non-equivalence** The problem is **not solved** and has **non-equivalences**
8		The student begins to move the cursor with: **Right** after **1.00 second(s)** for the next **3 turns** in **0.3 second(s)**.
11		The student deletes with: **BackSpace** after **0.80 second(s)**. 9x-(9+2x8) The step has a **non-equivalence** The problem is **not solved** and has **non-equivalences**

Figure 2. A sample of the replay clip.

IV. DISCUSSION

A. *Feedback from our Experts*

As our experts went about classifying our clips, we documented some of the more relevant conversation that occurred during the process. From this, we picked up some insight on the train of thought our experts went through in labeling off-task behavior.

1) *Identifying "Thinking"*

During experimentation, our experts found that one of the biggest determinants of off-task behavior was whether students' actions correspond to the proper way of finding the solution. If the numbers that the student typed reasonably resembled a number that was expected to come out given a problem, they would deem the student to be thinking about the lesson and thus be on-task. One problem in discernment using this method was that it was difficult for us to replicate this way of thinking operationally without having our model solve the problem each time. Not only was it difficult to determine whether the student was following a valid problem solving path, it was also difficult to determine if the student is also merely being careless (in these cases, our experts considered students to be on-task but confused). It was also insufficient to detect whether the students had partially solved the problem, which Aplusix provided as feedback. Later

in this paper, we introduce a "progression score" that we used to capture our experts' definition of "thinking".

2) Interface-related slips

Many students were found to be on-task and not-confused up to the point where they arrived at the solution. However, they were unable to "declare" the problem as "solved" using Aplusix's convention. When this happened, students performed random actions such as deleting their answer, perhaps thinking that it was wrong, or that the problem was not completely simplified. It may be argued that students should be classified as off-task at this point due to the nature of their actions. However, our experts considered these students to be truly at a loss because of the Aplusix interface. Therefore, students exhibiting these behaviors were regarded as on-task but confused.

3) Time as a Factor

Mrs. Arespacochaga said that one of the main factors she used in determining on-task behavior was time. If a student paused at the start of the exercise, the student is regarded as "thinking" but if a student paused at the end then the student was found to be confused and off-task. Upon further analysis, we found that majority of the clips classified as on-task resulted from three main attributes, two of which were time-related:

-The average time of each action across all actions performed was greater than 0.45 seconds.

-The total time of actions before the student becomes inactive for the rest of the 20-second clip is greater than 10.7 seconds.

These first two features reflected Mrs. Arespacochaga's thought-process of looking at when a student pauses. If a student paused at the end, it lessened their actions taken within 20 seconds and thus reduced action time and students who paused at the start and generally raised the average time across all actions taken.

B. Building the Model

As mentioned in the methodology, we now discuss the process of building our model. From what our experts said, the main criterion that determines whether a student is on-task was "thinking". The experts examined whether the numbers that the students typed were correct or if they were wrong due to carelessness. We attempted to perform this operationally using string parsing and simple logic.

In attempting to simulate our experts' criteria for *thinking*, we parsed each equation, starting from the original problem and identified the numbers present, whether they were coefficients, addends, or factors, and so on. For this explanation, let us use the example:

$$4(-x-7)+2(-9x+4)+6.2(3x+9)$$

Using our algorithm, we were able to identify the numbers: 4, 7, 2, 9, 6.2, and 3. Our parser was designed to get unique positive numbers. Negative numbers were treated as positive for our purposes and fractions were only known for their separate numbers as well. Since these set of numbers belonged to the original problem, they were given a score of 1, which means the student is not necessarily progressing if these numbers come up during the solution.

Our algorithm then performed simple arithmetic operations, namely addition, subtraction, multiplication, and division, with each permutation of numbers. This yielded a list of predicted numbers. A sample of these predicted numbers would be: 11, from 4 + 7; 28, from 4 * 7; 5, from 9 - 4; and a 6 again, from 4 + 2. These numbers were given a score of 5 signifying that if any of these numbers appear, a student has made some progress in solving the problem. Take note that we did not actually compute for the solution based on the original problem. We simply tried to perform possible operations that can occur between numbers to take into consideration careless mistakes made by the students in performing the wrong operation.

During the process of reading the clip line by line, the algorithm compared the numbers that appeared in the clip with the algorithm's list of numbers. If a similar number is found, the progression score is incremented with the score assigned to specified number on the list. That number's score will then be set to 0, stating that the predicted number has been found. If the number in the clip was not found on the algorithm's list, the new number will be added to the list and is given a score of 0. The algorithm then generated a new set of predicted numbers, incrementing existing ones by 1 and creating new ones with a score of 3.

The progression score was a quantitative measure for how reasonable the student's solution is based on the original problem and the successive steps taken by the student. In the case of clips that are found to be in the middle of the exercise, it emphasized the usage of predicted numbers rather than dwelling on the original problem. The scores were assigned to emphasize the importance of predicted numbers based on the original problem, over the predicted numbers based on the previous equations typed by the student. To further explain the relevance of this score, during our sessions with our experts, one of their main points in determining on-task behavior was when they find a number relevant to the original problem, they would deem the student on-task. For example, if the original problem was:

$$6x + 3(5x - 4)$$

and the student answered:

$$6x + 8x - 4$$

the experts regarded the student to be on-task and they would just write him / her off as being careless. On a similar note, if a student carelessly typed 18 and later corrected it with 16, which is part of the correct answer, the student is still considered on-task from the moment they typed 18.

Originally, we believed that Aplusix' feedback of equivalence in between steps would be informative enough to help us distinguish off-task behavior. However, because of the numerous instances of carelessness from our clips we found it necessary to create a feature that could capture those instances.

From the information we have gathered through our sessions with our experts, we were now ready to generate our models based on their classifications. The results of these will be discussed in the next section.

V. RESULTS

We manually removed some features based on our experts' feedback and then used the attribute selection algorithm provided by WEKA to reduce the feature space. We shall briefly discuss our sample sets and its characteristics based on the statistics of its features. We will then present the models we generated using WEKA[3].

[3] http://www.cs.waikato.ac.nz/ml/weka/

A. Features Used and Describing our Sample Population

In implementing our model based on the feedback from our experts, we extracted information from the clips as our experts analyzed them. The features used for our machine learning are as follows:

1) Problem difficulty and Complexity

One of the more basic features required by our experts was what type of problem and how difficult the student was trying to solve. This is usually the bases on how "reasonable" the pauses the student made were. Problem difficulty alone was not sufficient since more than 80% of the problems were of B1 – Expansion and Simplification. Problem complexity gives a numerical rating on how complicated the original problem appears to be as to possibly confuse the student. Because of this, our data represented behavior students who were answering relatively simple problems.

2) Starting Turn

Clips did not necessarily begin at the start of the exercise. They sometimes contained actions from the middle of problem solving process. Where in the process the student is plus the problem difficulty determined how reasonable the actions of the students are.

The majority of our sample population was composed of clips that started after students completed 44 or actions or less. This meant that students generally solved exercises without making too many unnecessary actions. There were a few exceptions, though. Some clips started after the student performed 575 actions.

3) Action Count and Time

Action Count indicated how many actions the student performed within clip. *Time* indicated the time from the first action to the last action.

From the statistics, the majority of our clips show that students were active throughout the 20-second intervals. The graph we see on Figure 3 shows that students performed as many as 55 actions within 20 seconds. As the statistics for action count show, the graph is skewed to the left and that as many as 100 clips would only contain 7 – 8 actions. This means that most of the students' actions performed were at a mean of 5.5 – 10 actions of the 20 second time window.

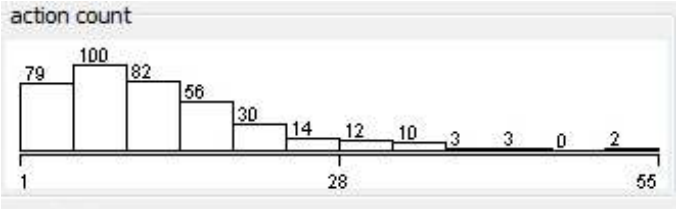

Figure 3. The left skew of the action count means that generally students did few actions within 20 seconds.

4) Deletion

To determine whether students were engaging in trial-and-error, we kept track of the deletion activity the students made and the other actions in between deletions. We believed that students who solved problems by trial-and-error would have bursts of deletion with little activity in between bursts.

We later had to refine the way we operationalized trial-and-error to consider the average action time, the progression feature and the cursor movements. Trial-and-error was usually performed quickly since the student had a set pattern of inputs i.e. consecutive numbers and usually by changing the same number [1]. Furthermore, trial-and-error resulted in a low progression feature since many of the numbers entered be unrelated to the numbers in the problem. Finally, cursor movement implied that the student is editing the equation part by part—a legitimate strategy in Aplusix. Cursor movements implied a thinking process as opposed to trial-and-error.

5) Keyboard Inputs and Interaction

This set of features constituted the number of each type of input the student made during the exercise. These inputs included number inputs, symbol inputs, letter inputs, cursor movement, editing functions such as cut and paste, comments made by students, and miscellaneous functions such as declaration of problem solved or abandonment.

6) Solution status

Solution status refered to the correctness or wrongness of the student's solutions. We kept track of the number of help requests made, if the solution was abandoned, whether the student able to solve it wholly or or partly, whether each step was equivalent to the other, and how many steps the student executed within the time span of the clip.

Many of the clips did not capture the end part of an exercise where a student either solved the problem or abandoned it. This was evident from the fact that there was only one instance of abandonment and 22 instances of a clip ending in being solved. There were, however, numerous clips found to be "partly solved". This meant that the student was able to solve the problem but was not able to declare it solved, probably because there may have been still some non-equivalence in some of the steps.

7) Progression

This feature was the score rating of each equation after each action of the student based on the relationships of the numbers found within to the original problem. As explained earlier, this feature was derived from an algorithm we created in order to simulate the thought-process of our experts to take notice of the numbers the students are working with in relation to the original problem. The clips were given a higher progression score if the students were found to be working on predicted numbers, which were based on original numbers with some sort of mathematical operation performed on them, rather than numbers that had no relation to the original problem or the original numbers themselves.

A. Ms. Arespacochaga's Model

Out of the 391 samples, our expert, Mrs. Arespacochaga classified 283 clips as being on-task, 80 as off-task, and 28 as unknown. With this model, the algorithm is able to classify 80% of our samples correctly and after performing a 10-fold cross-validation, we got a result of a Kappa statistic of 0.4848. Figure 5 shows the decision tree for the model generated.

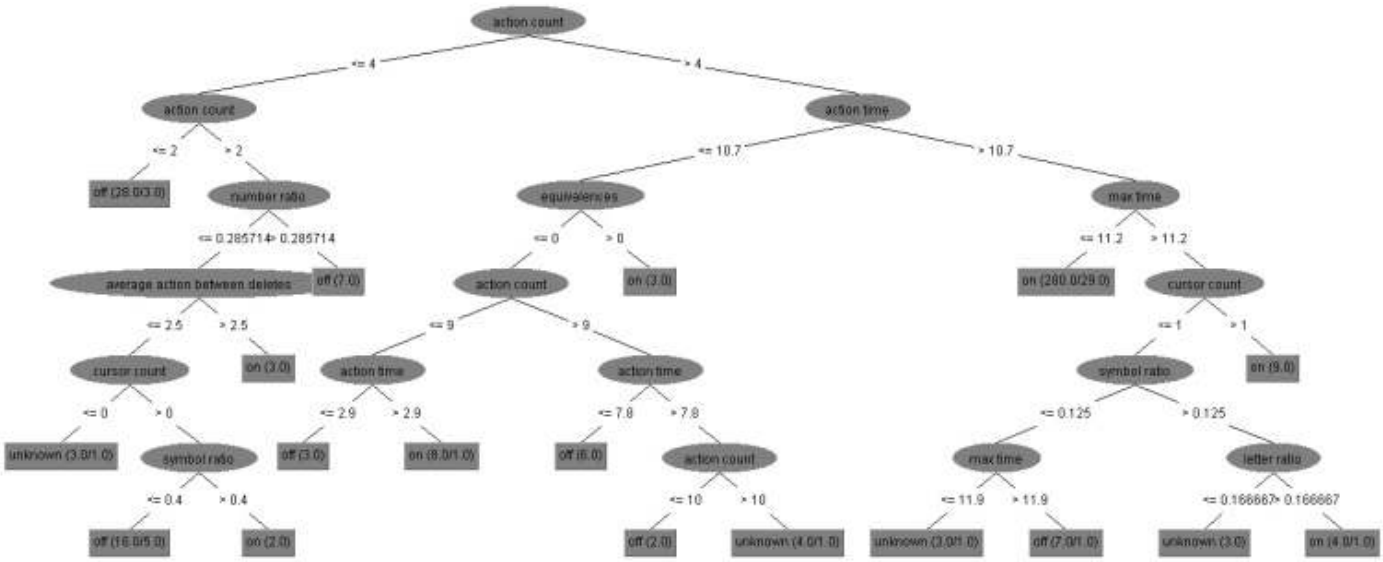

Figure 5. A tree generated from Arespacochaga's labeled clips starts with action count as its first criteria in decision-making.

B. Dr. Soto's Model

Out of the 391 samples, our expert, Dr. Soto classified 319 clips as being on-task, and 72 as off-task. With this model, the algorithm is able to classify 87.7% of our samples correctly and after performing a 10-fold cross-validation, we got a result of a Kappa statistic of 0.5477. Figure 6 shows the decision tree for the model generated.

Both models use action count as its first criteria in the decision-making process and close to it would be action time. These results may reflect what Arespacochaga mentioned during classification about checking if the student pauses early or late. It is possible that short clips tend to provide insufficient information on behavior such that the few random actions that is done within the 20 seconds could already give a sign that the student is confused with the work and tend to become off-task.

VI. FUTURE WORK

The models created generated a moderate value of Kappa, giving us satisfactory results that it is possible to automatically detect off-task behavior using text-action logs from an intelligent tutor for Algebra. If we were to improve the results of the experiment, a research that focuses on the features themselves could further refine the model generated from the algorithm. Our features were derived from the feedback of two experts. It is possible that opinions of other people will open up other ideas on the features we can extract from action clips. Although we believe that our sample set was sufficient as a representation of the whole population of clips, it is also possible that classifying more clips could level out any inconsistencies our experts may have accidentally shown in these few 391 clips and thus provide WEKA with more information to work with and further increase the reliability of the model.

The current work has several limitations. First, there was one model created per expert as opposed to one model that combined the experts' opinions. Separate models were created because agreement between raters was low (Kappa = 0.49). Another set of clips was generated and relabeled by the same experts in an attempt to arrive at greater agreement. However Kappa turned out to be even lower at 0.30. Second, the model was created off-line, with pre-recorded logs. It was not integrated with Aplusix. Future work can attempt to come to greater agreement between raters and to produce model that consolidates the experts' opinions. Appropriate interventions can be designed based on pedagogically sound practices. The model and these interventions can then be integrated into Aplusix.

ACKNOWLEDGMENT

We would like to thank Dr. Cornelia Soto and Mrs. Ria Arespacochaga for their cooperation in this research as our experts. We would also like to thank the Department of Science and Technology's

Engineering Research and Development for Technology Program for the grant entitled, "Multidimensional Analysis of User Machine Interteractions Towards to the Development of Models of Affect."

REFERENCES

[1] Baker, R.S. Corbett, A.T., Koedinger, K.R., and A.Z. Wagner (2004), "Off-task behavior in the cognitive tutor classroom: when students 'game the system'", Proceedings of ACM CHI 2004: Computer-Human Interaction 383-390

[2] Baker, R., Corbett, A. T., and A.Z. Wagner (2006), "Human classification of low-fidelity replays of student actions", Proceedings of the Workshop on Educational Data Mining, Jhongli, Taiwan, pp.29-36.

[3] Hulshof, C. D. (2004), "Log file analysis", Encyclopedia of Social Measurement

[4] Koedinger, K.R., Anderson, J.R., Hadley, .W.H., And M.A. Mark (1997), "Intelligent tutoring goes to school in the big city", International Journal of Artificial Intelligence in Education, 8, 30-43

[5] Murray, Tom (1999), "Authoring intelligent tutoring systems: an analysis of the state of the art" International Journal of Artificial Intelligence in Education, 10, 98-129

[6] Rodrigo, Ma. Mercedes T., Ryan S.J.D. Baker, Sidney D'mello, Ma. Celeste T. Gonzalez, Maria C.V. Lagud, Sheryl A.L. Lim, Alexis F. Macapanpan, Sheila, A.M.S. Pascua, Jerry Q. Santillano, Jessica O. Sugay, Sinath Tep, and Norma J.B. Viehland (2008) "Comparing learners' affect while using an intelligent tutoring system and a simulation problem solving game", Proceedings of the 9th International Conference on Intelligent Tutoring Systems, pp 40-49

[7] Rowe, Jonathan P., McQuiggan, Scott W., Robison, Jennifer L. (2009), and James C. Lester, "Off-task behavior in narrative-centered learning environments"

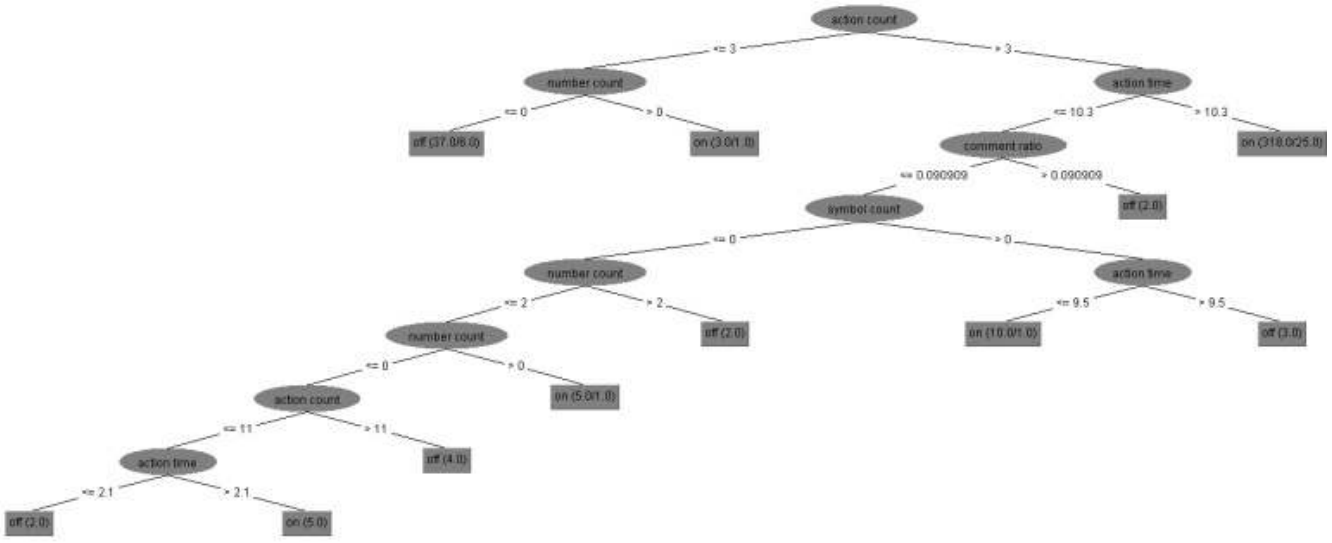

Figure 6. A tree generated from Dr. Soto's labeled clips starts with action count as its first criteria in decision-making.