Use of Machine Learning Classification Techniques to Detect Atypical Behavior in Medical Applications

Terrence Ziemniak, CISSP
Director, IS Security and Compliance
Resurrection Health Care
Chicago, IL USA
TMZiemniak@Yahoo.Com

I. Introduction

Health care informatics is growing at an incredible pace. Originally, health care organizations, like all other industries, used pen and paper to track medical information. Ten years ago the more mature health care organizations had simply practice management applications. Today, these organizations have full blown electronic health records systems. Tomorrow these organizations will be sharing information across the globe.

Physicians (and the sponsoring organizations) are obligated to protect this data. Health care has followed the trend of many other industries in implementing technologies and processes to address certain risks. Encryption is enabled to ensure confidentiality. Business continuity techniques are applied to ensure system availability. However, there is no best practice solution that can be applied to the problem of detecting inappropriate activity. How can a hospital tell when Nurse Smith is "snooping" in medical records? How can a radiologist tell when a lab technician is feeding information to a law firm?

This paper describes a system that will detect atypical behavior in a health care application. The first section will discuss the impetus for such a system. The second section will describe the design and implementation of this system. The third section will document a series of experiments showing the effectiveness of detecting atypical behavior and the analysis of whether said behavior was inappropriate.

II. Background

What I may see or hear in the course of treatment or even outside of the treatment in regard to the life of men, which on no account one must spread abroad, I will keep myself holding such things shameful to be spoken about[1].

Physicians and health care organizations are privy to a large amount of private information. As custodians of this data they are obligated to ensure its privacy and security. In the days of paper records this was a manageable exercise with minimal risk. Physical locks kept people away from records.

But with the advent of electronic medical records the landscape has changed. In a large modern health care system, the intensive care unit will share the same data store as behavioral health (though hopefully with different access rights). Records are extremely easy to copy and move. Overlay this with the enormous complexity of modern health care business; for example, with employees, contractors, students, job sharing, outsourcing, and the mobile workforce ensuring the privacy of medical information is an extremely difficult task. Guaranteeing privacy is impossible.

Luckily the information security community has reached the level of maturity where people understand that security and privacy might never be guaranteed. Organizations generally strive for reasonable security by means of various controls, such as policies and robust access control. One of the most fundamental controls is auditing whereby activity is recorded. This allows for "reasonable and appropriate[2]" controls to enforce security with the ability to look and see if these were circumvented or misused.

Most modern health care applications provide robust auditing. However there is no easy way to detect inappropriate activity using static rules. Consider the following scenarios:

[1] Hippocratic Oath

[2] This language is directly pulled from the HIPAA security ruleset.

978-0-7695-4403-8/11 $26.00 © 2011 IEEE
DOI 10.1109/IMF.2011.20

- Is it inappropriate that Nurse Smith looked at 20 patients?
- Is it inappropriate that Nurse Smith printed 50 documents?
- Is it inappropriate that Nurse Smith signed on to 10 workstations?

Due to the variable nature of healthcare workflows, these may be either appropriate or not. But what if it were possible to compare Nurse Smith's activity to that of other nurses? If Nurse Smith's activity was significantly different than his peers then this might warrant review.

III. CHALLENGE

With the obvious need to look for unauthorized access to protected information, why is this so difficult? This appears to be a simple question of who needs to see what information. However, for a variety of reasons there is no easy answer to this question.

The main issue is that medical entities have typically defaulted to open access instead of closed access. This is due to the nature of the work being done. In health care quick access to data is of paramount importance. Imagine the case where a physician is prevented access to critical data in the middle of a procedure.

There are also concerns about the efficiencies of physicians. Physicians will view any technology or process seen as impeding a physician work unfavorably. Health care organizations are keenly aware of this and account for this when designing systems. It would be unacceptable to have physicians call the help desk to get access to a patient's information.

The last factor is the complex web of relationships in health care. This creates a very convoluted list of requirements that generally start from the basis that a physician has access to his patients in the system in which he works. Physician staff derives rights to the same patient data based on their relationship to the physician.

This is extremely difficult to implement effectively. Consider the following complexities:

- A medical coder will need access to most sensitive information regarding a client for billing purposes.
- Outsource clinical services exist for reading of medical images.
- Physicians may or may not be employed by a hospital.
- Physicians may have relationships with multiple hospitals.
- Nurses often need access to all information that their doctor can access.
- Non-clinicians (such as dieticians) may need to access parts of the patient's information.
- Information on a given patient may exist in multiple locations such as in a hospital's electronic medical record system (EMR), a practice's practice management system (PMS) and in his personal health record (PHR).
- Non-clinical services such as billing and transcription are often outsourced.
- Radiologists are not directly associated with any patient but would need to access many records.

Now consider the case where Nurse Smith is snooping on his neighbor's records at the hospital. How can this be detected? It is a very difficult situation that cannot be addressed via a simple rule-based analysis. The biller looking at the most number of patients is not necessarily being malicious. The pharmacist that looks at more than 25 patients is not necessarily being malicious. A static, rule-based approach to detecting atypical behavior is not effective.

However, what if it could be flagged when Nurse Smith, the biller and the pharmacist were not acting like their peers? What if it were possible to map out how people in a given role acted and then flagged those who fall outside of that norm? It would then no longer be necessary to define a list of inappropriate activity with static thresholds. Instead it could be asked, "Who is acting in an unexpected manner?"

Luckily this problem is not unique to health care informatics. There is a whole branch of computer science called Machine Learning (ML) that designs tools which automatically detect patterns and derive meaning from them. Machine learning provides a means of building models of typical behavior and identifying the misclassified behavior/items thereby allowing detection of atypical behavior. Machine learning techniques have been successfully applied to many common information security problems [2] such as spam filtering [6] and virus protection [5].

IV. Hypothesis

The following describes a system that will use classification models to identify which users of a medical application are not acting like their peers (i.e. atypical). This method of anomaly detection will effectively flag this behavior without the need for a rigid rule-based analysis.

The paper will attempt to show that the machine learning classification based on user's roles followed by prescribed investigation techniques produce a significant quantity of truly suspicious users.

V. Data

Data for this experiment was pulled from a medical application, McKesson Portal. This medical application is the primary tool used to view data from multiple in-house medical applications. Logs representing two months of activity (October and November 2009) were included. The health care system from which these were pulled consisted of 9 hospitals and 15,000 employees. The employees were in one of many possible roles such as physician, pharmacist or social worker. The three roles with the most users were physicians (58%), physician assistants (18%) or nurses (5%).

The data used (i.e. application logs) did not contain any medical information but did contain information that would be considered confidential (patient names). It was not necessary to put this confidential data into the database.

VI. System Components

A. PreProcess

Each log entry represents a single transaction for a user. For the purpose of analyzing user activity, it was necessary to consolidate data into periods of time. A single transaction of Dr Smith logging into the system is a poor data point. However by viewing all transactions for Dr Smith for a given day, this can be more readily compared and contrasted with other users of the system.

- How many times this day did Dr Smith sign in?
- How many patients did Dr Smith view?
- How many workstations did Dr Smith use this day?
- How does this compare to other physicians?
- How does this compare to nurses? Or to receptionists? Or to other staff?

The raw logs were processed into 24-hour blocks. This means that for the purposes of classification, an instance is the cumulative activity for a user for a single day. These instances are then imported into a database for use by the classification system as well as for use in subsequent investigations.

There are three general types of data stored for each instance. First there were many simple counts. For example an instance shows the total number of patients seen by the user for that day.

Secondly the system recorded aggregate information (across the entire 24 hour period) for an instance. This aggregate information can be significantly different from the simple totals. For example a nurse working in intensive care unit may have worked with 10 patients looking at each patient's records 5 times. That would be recorded as 50 total patients but only 10 unique patients.

Lastly, in addition to recording total and aggregate counts, other data was added regarding the user. Specifically it was necessary to record supplemental information to help classify the user. For example the raw log files record the user name, such as ASmith, but not necessarily the role. To analyze activity from a role-based perspective, it was necessary to tell if ASmith is a physician, a nurse or something else.

B. Data Structure

After processing the application log files, the following data points were recorded for each instance. Note that an instance represents a day's worth of activity for a given user. These are the data points that will be analyzed to by the classification engine.

Column	Description
User	Name of user

Logins	Number of logins by this user
Hours	Number of hours worked.
LogOuts	Number of times user actively logged out of the system
Suspends	Number of times user was suspended due to inactivity
ModulesUnique	Number of unique modules (screen views) accessed by the user
ModulesTotal,	Total number of modules (screen views) accessed by the user
PatientsUnique	Number of unique patients accessed by the user
PatientsTotal	Total number of patients accessed by the user
APWW	Access Patient with Warning. This 'break the glass' event is triggered when user attempts to access a patient where an explicit (e.g. patient's primary physician) or inferred relationship (e.g. nurse assigned to patient's ward) does not exist.
PrintModule	Number of print jobs performed by user
WorkstationsUnique	Number of unique workstations used by user

C. Classification System

The WEKA[3] data mining software package was used as a classification engine. This tool is a multipurpose machine learning engine that supports many types of classifications based on common machine learning algorithms.

For the purposes of this exercise, a single classification algorithm was used. The J48 algorithm in WEKA produces a C4.5 decision tree. This type of classification was chosen for two reasons. First this model scales to large datasets – in the examples detailed below, the system is processing tens of thousands of instances. Secondly a decisions tree has the benefit of producing output that can be easily interpreted by a user and it could be easily understood why a given instance was misclassified [4].

In some cases parameters were used to adjust the level of pruning associated with the decision tree. The *confidence level* parameter in J48 affects the tree size by setting the threshold for how confident the system should be in the decision. The *minimum number of* objects parameter controls the minimum number of objects that should be in a resulting leaf.

WEKA will construct a decision tree, which most accurately classifies all instances on a given attribute such as role (i.e. physician or nurse). Once the tree is generated the system runs the original data set through the tree to determine accuracy. There are several metrics produced that measure the accuracy but two are especially important for our purposes. First, the overall accuracy is recorded showing how accurate the tree classified all users.

Second, there is a separate accuracy rate for the users in the specific role on which the tree is attempting to classify. For example, if a tree attempts to classify physicians and has an overall accuracy rate of 80%, this means that four out of five users (physicians and non-physicians) were correctly classified. Additionally, the system's ability to classify physicians might be 90%. This says that if someone were a physician, the system would correctly classify him or her nine out of ten times. This was important in cases where the role in question represented a small number of the total user base.

The third accuracy measurement is associated with a given leaf of a tree. For each leaf the accuracy is calculated for all instances that navigated to this particular node.

VII. EXERCISES

The remainder of this paper will document various attempts at using the above described system to find atypical behavior, investigate it and determine if this behavior represents inappropriate activity. The efforts documented below start with a description of each run. This describes the starting data set and the initial classification run. Subsequent to each run is a series of investigations. These are attempts to derive meaning from the results of the classification run.

A. Classification Runs

The exercises described below are divided into a series of classification runs. Each run will process a subset of the entire WEKA database. For example, a classification run will likely want to limit the site, date range and attributes for consideration. This last component is a very important consideration as it designates which of the many attributes available for each instance will be used when creating the classification tree. It has been show [1], [7], [8] that proper filtering will increase the accuracy and efficiency of the tree building exercise. This attribute filtering was done via a combination of automated filtering techniques as well as manual review.

B. Types of Investigations

When reviewing the results of a given classification run, investigations are done in one of several ways. The first method uses ad-hoc analysis to find atypical behavior. This is generally done by visual inspection looking for interesting nodes. Attributes of a node that could trigger investigations include classification accuracy (or inaccuracy) and the nature of the path (e.g. large number of unique patients).

The second method of investigation takes into consideration the length of the path to a given node. How does a node that is reached by a single decision compare to one that takes 8 decisions? For these investigations, samples will be taken that represent short, medium and long path lengths.

The third method attempts to make a more accurate tree by removing outlying instances. This filtered-classification is done by running a normal initial classification run and then pulling out all misclassified instances. After this the remaining dataset is reclassified. The resulting decision tree is then analyzed by either ad-hoc or path-length investigation.

It should be noted that in the investigation of misclassified instances, both false-positive and false negative classifications were reviewed. The specific meanings of these cases are relative to the specific run. In the case of classifying physicians, an example of a false positive would be the case where a nurse is incorrectly classified as a physician, and an example false negative would be the case where a physician is classified as a non-physician. Either case can be troubling. The first could indicate a snooping nurse while the second may indicate the doctor has shared his credentials. Both are cases of inappropriate activity.

C. Node Research

Once it is decided that a node is to be investigated, two primary tools are used for research. The first method consists of manual review of the raw log files. This will show specific detailed information.

The second method of researching nodes is by using SQL calls against the WEKA database. This produces a simple high-level view of the misclassified (or properly classified) instances. Additionally this can give a view into the historical behavior of any user across the entire dataset. This is a valuable tool when determining whether or not atypical behavior (misclassified) is inappropriate activity.

To show how historical information can help when investigating instances, consider the case of a misclassified nurse. If Nurse Jones had a misclassified instance where in a 16 hours shift she looked at 100 unique patients, this in itself may be suspicious. However it might be that in a typical day Nurse Jones looks at 50 unique patients. Furthermore it might be that Nurse Jones' typical shift is 8 hours, so this is in line with her typical historical behavior. So in the case described above the higher-than-normal values might simply indicate that Nurse Jones has worked a double shift that day.

D. Summary of Runs

Below is a list of classification runs detailed in the next section. For each run detailed output from WEKA and SQL are available in the supporting documents section.

- Run 38 classifies residents at site 4. Ad-hoc investigations discovered several users with unusual activity.
- Run 46 combines residents and physicians at site 4 in a single classification group. The path-length investigations produced multiple unusual instances.
- Run 42 investigates physician assistants from site 0. The filtered-classification investigations produced multiple suspicious instances.

E. Run 38

This run analyzed residents (i.e. physicians in training) from site 4. The following SQL code was used for the date import. Note that this classification run utilized six attributes: hours, number of unique modules, number of total modules, total number of patients, total number of times residents' access patient with warning[3] and number of modules printed.

```
SELECT hours, modulesunique, modulestotal, patientstotal, APWW, PrintModule, WorkstationsUnique,
if(rolead = 2, '1', '0') as class
```

[3] APWW event is recorded when a user accesses a patient's information when there is no documented relationship (within the application) between the user and the patient. This is not an uncommon event as it would be extremely difficult to document the many possible relationships. For example radiologists will get APWW when they access any patient.

```
from instances, users
where instances.user = users.user
and site = 4
```

The J48 classifier (with parameters of confidenceFactor of .05 and minNumObj of 5) produced an overall accuracy of 82.3% and classified residents with an accuracy of 77.5%. The following tree was produced.

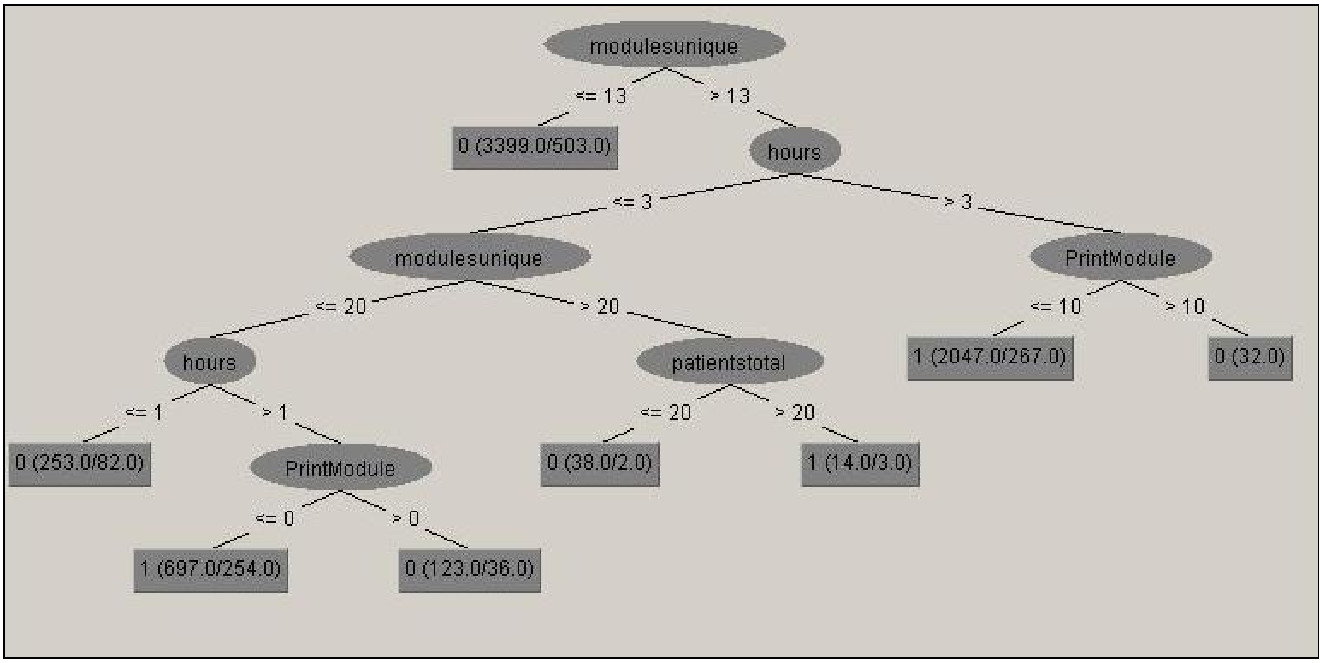

Figure 1

For this run investigations were run in an ad-hoc method; that is they were based on the analyst's intuition to determine which nodes should be reviewed.

1) Investigation 1

There is a leaf in the middle [0 (38.0/2.0)] where 38 instances ended up. The system predicted that these users were not residents (i.e. class 0). The classification was accurate for 36 out of 38 instances (94.7%). Using the following SQL code, the two users were identified.

```
SELECT * FROM weka.instances i, weka.users u
where modulesunique > 20
and hours <= 3
and patientstotal <= 20
and site = 4
and rolead = 2
and i.user = u.user
```

The mapping shows that these two residents did in fact act unusually as compared to their peers. But did they act in ways inconsistent with their own historical usage patterns? Analysis of the WEKA database allowed us to make the following observations.

- A user had 50 recorded instances over the entire data set. The misclassified instances were consistent with this historical behavior.
- A user had 29 recorded instances over the entire data set. His usage across all metrics was considerably less than average including his number of hours. This would indicate that this resident was acting normally, but just worked a shorter day.

2) *Investigation 2*

There is another leaf in the middle [1 (14.0/3.0)] where 14 instances ended up. The system predicted that these users were residents. The classification was accurate 11 out of 14 times (78.5%). The following SQL call was used to show the misclassified instances.

```
SELECT * FROM weka.instances i, weka.users u
where modulesunique > 20
and hours <= 3
and patientstotal > 20
and site = 4
and rolead <> 2
and i.user = u.user
```

Activity from the 3 misclassified users was reviewed.

- A user had 54 instances over the entire dataset. The user's activity was consistent with his historical patterns.
- A user had 51 instances over the entire dataset. The user's activity was consistent with his historical patterns.
- A user had 37 instances over the entire data set. While working an expected number of hours, this user saw considerably more patients and more modules than normal. To further investigate this user, the following steps were taken.
    - o Upon further review, it was determined that the user was misclassified and was actually a physician.
    - o The number of unique modules, 27, substantially deviated from his historical activity.
    - o However the number of patients did not deviate. Upon review of the modules accessed by this user on the atypical day, there is no evidence of inappropriate activity. This is based on the function of the given modules.

3) *Investigation 3*

There is a node [0 (253/82))] where 82 residents were misclassified. The system predicted that these users were not residents. The classification had a relatively low accuracy rate of 82 out of 171 times (47.9%).

```
SELECT * FROM weka.instances i, weka.users u
where site = 4
and rolead = 2
and i.user = u.user
and modulesunique > 13
and modulesunique <= 20
and hours <= 1
```

The following observations were made regarding this node.

- One user was responsible for a misclassified instance where his patient and module count were higher than normal but his hours worked were much lower than normal. The pattern of activity (access patient and then review demographics and test results) matches his historical behavior so this was not considered inappropriate.

F. Run 46

This run analyzes the combined role of physicians and residents (physicians in training) from site 4. The following SQL code was used for the date import.

```
SELECT modulesunique,  patientstotal, APWW,  printmodule,
if(rolead = 2, '1', if(rolead = 1, '1', '0')) as class
from instances, users
where instances.user = users.user
and site = 4
```

The J48 classifier (with parameters of confidenceFactor of .05 and minNumObj of 5) produced an overall accuracy of 87.1% and the true positive classification of physicians and residents with an accuracy of 98.2%. The following tree was produced.

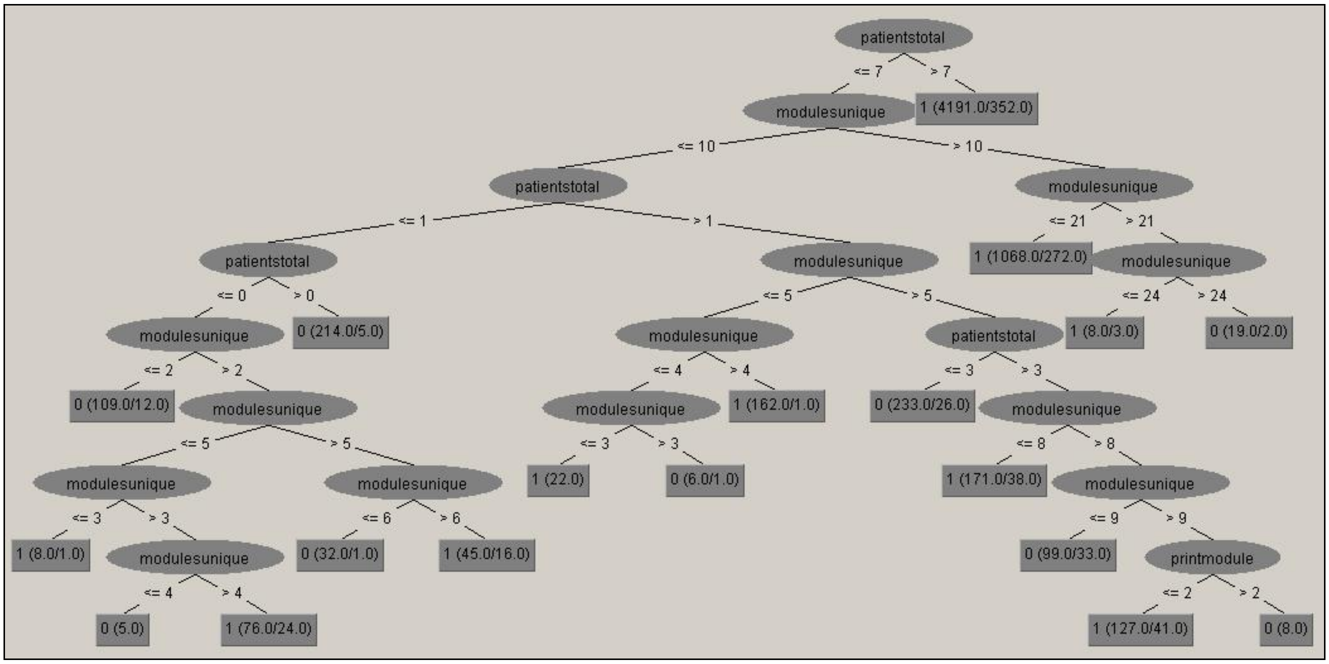

Figure 2

This run was selected to see whether or not the location of the node in the tree had any bearing on its usefulness. The different runs produce trees of various depths. What does it mean that a group of instances are classified based on a single attribute? What does it mean when a group of instances require nine decision points to be classified? The following investigations will compare nodes at different path lengths to see if the number of decision points would help improve the usefulness of the tool for finding inappropriate behavior.

1) Investigation 1

The first investigation deals with nodes with short paths. The node [1(4191/352)] correctly classified the combined role 3839 times with an accuracy of 91.7%. The following SQL call was used to show the misclassified instances.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 4
and patientstotal > 7
and rolead not in (1, 2)
```

Of these 352 misclassified instances these observations were made.

- A user had a very large increase in the number of logins in this instance. His average number of logins was 12 but this instance he logged in 70 times. The hours are higher than normal (7 versus 4) but not enough to account for the large increase in logins. Upon further investigation it was found that this user was part of the application support team and was testing the system's single sign-on functionality.
- A user had a large increase in both the number of PatientsUnique and PatientsTotal when compared to his historical activity. Furthermore this user had a marked increase in the number of print jobs. It was found that this user has a special bi-monthly task that explains this behavior. This was considered normal activity for this user.

2) Investigation 2

This investigation deals with nodes with medium length paths. The first node [0(19/2)])] classifies users as not physicians or residents with an accuracy of 89.5%.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 4
and patientstotal <= 7
and modulesunique > 24
and rolead in (1, 2)
```

Of these 2 misclassified instances these observations were made.

- Both users had a large increase in the number of ModulesTotal when compared to their historical behavior however there was a proportionally similar increase in the number of ModulesUnique. Furthermore, there was no increase in either the number of patients or the number of hours. These two physicians were likely investigating new capabilities or modules in the system.

The second investigation of medium length paths involves node [1(162/1)])]. This node classifies the combined role with an accuracy of 99.4%.

```
SELECt * from instances i, users u
where i.user = u.user
and site = 4
and patientstotal > 1
and modulesunique = 5
and rolead not in (1, 2)
```

For the 1 misclassified instance these observation were made.

- This user's activity is in line with his historical behavior. During investigation it was noticed that this user's application RoleID indicated his role was that of a physician. But the credentials used for classifying (i.e. Windows account) were blank. It is likely that this user was correctly classified by WEKA but simply mislabeled in RoleAD.

3) Investigation 3

This investigation deals with nodes with longer length paths. The first node [1(127/41)] classifies the combined role, physicians and residents, with an accuracy of 67.8%.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 4
and rolead not in (1, 2)
and patientstotal in (4, 5, 6, 7)
and modulesunique = 10
and printmodule <= 2
```

Of these 41 misclassified instances these observations were made.

- None of these instances showed any substantial change from the user's historical patterns.
- Additionally there were no values which registered as suspicious.

The second investigation of longer length paths involves node [0(32/1)])]. This node classifies role 0 (i.e. not physicians or residents) with an accuracy of 96.8%.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 4
and rolead in (1, 2)
and patientstotal <= 0
and modulesunique = 6
```

For the 1 misclassified instance these observation were made.

- It was extremely unusual that the user was active in the system but accessed no users. This was in line with his 4 other instances in the 60 day dataset so no further investigation is warranted.

G. Run 42

This run analyzed physician assistants from site 0. Note that this time the role attribute was pulled from the application (RoleID) instead of from Windows (RoleAD). The following SQL code was used for the date import.

```
SELECT modulesunique, patientsunique, PrintModule, WorkstationsUnique,
if(roleid = 402, '1', '0') as class
from instances, users
where instances.user = users.user
and site = 0
```

This run did not use the typical J48 parameters. The original run produced a very large tree with 191 leaves and a total size of 381. To shrink down the tree, the J48 attribute governing the minimum number of objects per leaf was changed from 5 to 50. This had a small negative impact on the overall accuracy (82.3% to 77.3%). But the resulting tree was much more usable so this was considered acceptable.

The J48 classifier (with parameters of confidenceFactor of .05 and minNumObj of 50) was applied and produced an overall accuracy rate of 77.3%. The classification accuracy of actual physician assistants was 70.5%. The following tree was generated.

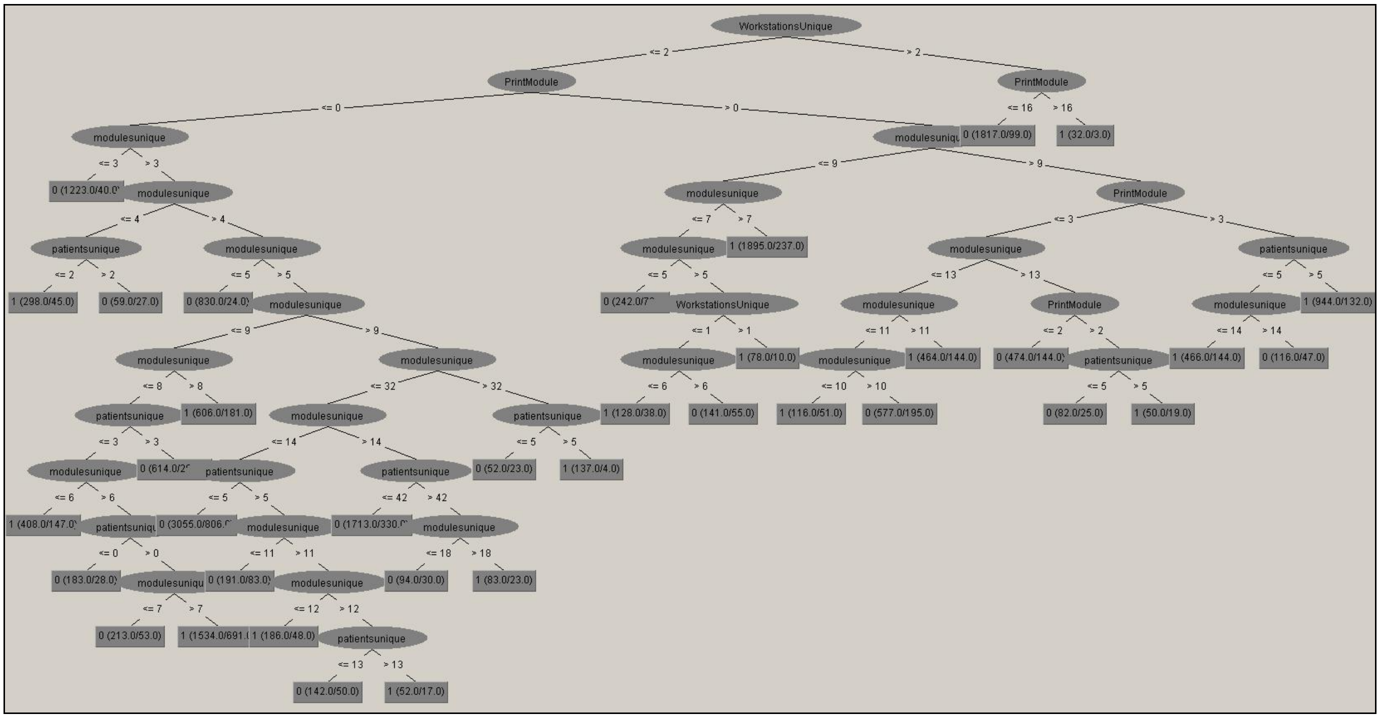

Figure 3

This third type of investigation differs substantially from the first two (ad-hoc and path length) in that a second tree is created. The intent is to pull out the outliers - those misclassified in the original classification tree - and then generate a new tree.

All instances that were misclassified in this run were then removed (4,647). This included both physician assistants that were misclassified as non- physician assistants as well as non- physician assistants misclassified as physician assistants. The data set was then reclassified by WEKA

The J48 classifier (with parameters of confidenceFactor of .05 and minNumObj of 50) was applied and produced an overall accuracy rate of 98.6%. The classification accuracy of actual physician assistants was 98.2%. The following tree was generated and used for the following investigations.

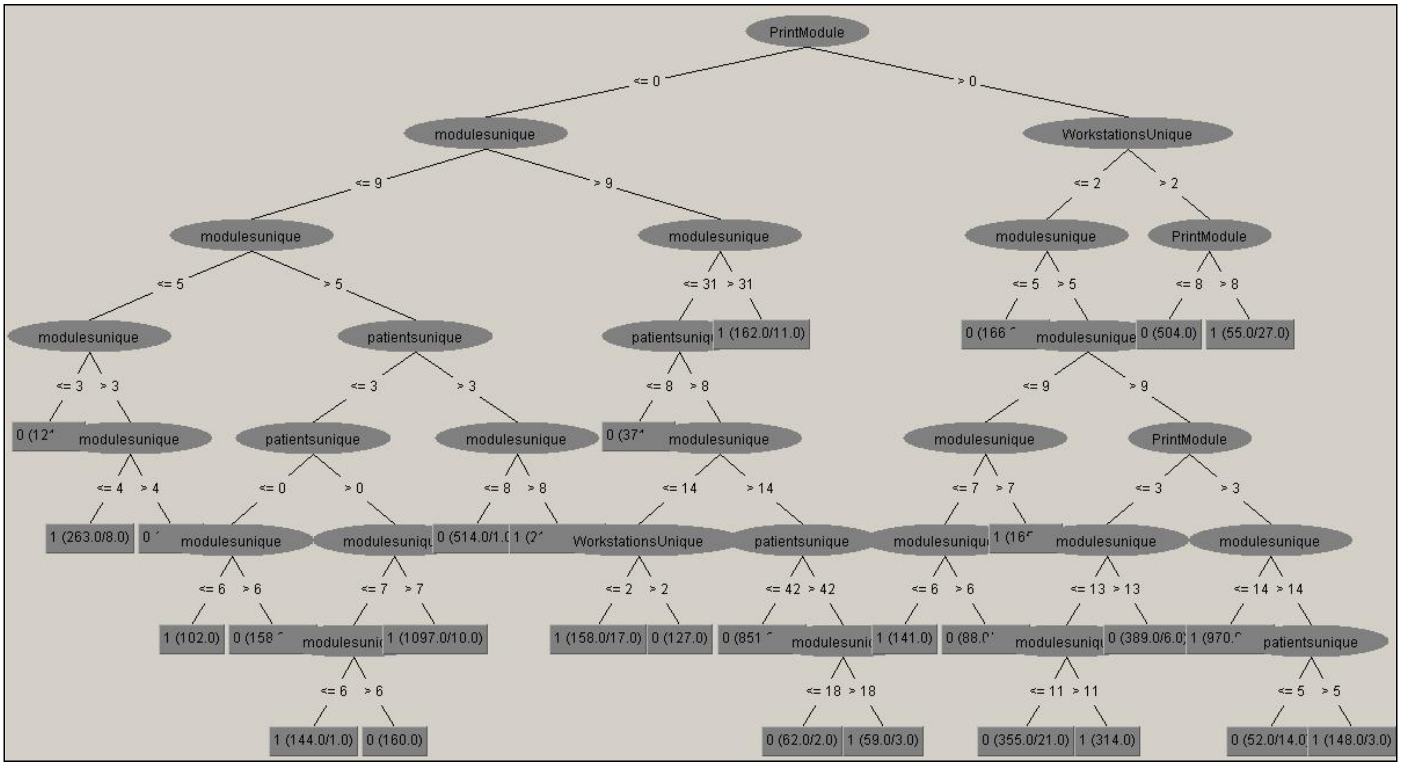

Figure 4

1) Investigation 1

The first investigation for a short path was node [1(162/11)]. This node contained 11 non-physician assistants that were misclassified and an overall accuracy of 93.2%.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 0
and ignoreinstance = 0
and roleid <> 402
and printmodule <= 0
and modulesunique > 31
```

The following observations were made regarding this node.

- Of these 11 instances, 10 were IS staff.
- The one other misclassified instance was from a physician resident. This user had a significantly more ModulesUnique (39) compared to his average (15). During research, it was found that this discrepancy was due to the user testing the mobile version of the application that day.

The second short-path node investigated was node [1(55/27)]. This node misclassified 27 non-physician assistants with an overall accuracy of 50.9%.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 0
and ignoreinstance = 0
and roleid <> 402
and printmodule > 8
and workstationsunique > 2
```

The following observations were made regarding this node.

- The 14 users were responsible for the 27 instances. Two individual had more than 4 misclassified instances in this node.
- One physician had 14 print modules even though her historical average was 0. These were facesheet reports and over a 10 minute period. This was likely done for billing purposes.

2) *Investigation 2*

This investigation examined a node with average length paths. The first investigation was node [1(263/8)]. This node misclassified 8 non-physician assistants with an overall accuracy of 96.9%.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 0
and ignoreinstance = 0
and roleid <> 402
and printmodule <= 0
and modulesunique = 4
```

The following observations were made regarding this node.

- One user accounted for 4 of these instances. In two of these instances most of his activity was less than normal including his hours. This user's activity over the full data set shows a wide range of values for most attributes.
- It is interesting to note that a previous discussed user has two instances in this node. During the filtering process to remove outliers, 7 of this user's instances were set to ignore; leaving 42 instances for this second pass. This would indicate inconsistent work habits for this user.

3) *Investigation 3*

This investigation examined nodes with longer length paths. The first node [1(148/3)] had 3 instances of non-physician assistants being misclassified with an overall accuracy of 97.9%.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 0
and ignoreinstance = 0
and roleid <> 402
and printmodule > 3
and modulesunique > 14
and workstationsunique <= 2
and patientsunique > 5
```

The following observations were made regarding this node.

- Two of the misclassified users were physicians.
- The other misclassified instance was due to a user with a job description of data assistant. This user's activity is consistent with his historical behavior aside from the fact that he printed four times when he averages 0. The four modules printed had no patient information so this is not considered suspicious.

The second node of longer paths was [0(355/21)]. This node contained 21 misclassified physician assistants with an overall accuracy of 94.0%.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 0
and ignoreinstance = 0
and roleid = 402
and printmodule in (1, 2, 3)
and modulesunique in (10, 11)
and workstationsunique <= 2
```

The following observations were made regarding this node.

- Activity for all 21 instances was at or below historical averages.
- When looking at RoleAD (which tends to be more accurate), 10 of the misclassified users were physicians.

4) Investigation Ad-hoc

This same run had been previously analyzed by means of a simple ad-hoc review. There were two significantly atypical instances that were found by ad-hoc review. These are true outliers and were filtered by design.

- A user printed modules 50 times while in his other 51 instances he averaged 1 printer per day. Similarly his patients total was 91 for this instance but averaged less than three for the other days. This was likely due to the physician's billing procedure.
- A user viewed 93 unique modules in a single instance in which he worked only 1 hour. The other two instances for this user, he averaged 11 unique modules. This ended up to be a bug in the system where activity due to a single module was incorrectly logged. However with the data available this certainly constituted atypical behavior and ideally would have been reviewed.

VIII. Conclusion

This paper described the design and use of a system that can be used in medical applications to help comply with the bevy of requirements that obligate them to look for inappropriate activity.

By implementing machine learning methodologies, the system was successful at detecting atypical behavior. It was able to accurately detect when a physician was not acting like a physician. Conversely it successfully detected when a pharmacist was acting like a physician.

All three investigation types produced instances that were suspicious. The ad-hoc appeared to be the most fruitful. This was likely due to the analyst?s intuitive review of the decision tree. However all types of investigations produced meaningful results.

Of the ten investigations documented in this paper there were many types of suspicious activity that were worthy of investigation. Recall that the activity listed below had already determined to be atypical when compared to his peers.

- A user with a large number of sign-ons - due to testing of a single sign-on solution.
- Users that access a large number of modules - due to training
- Users printing an unusual number of print jobs - due to billing functions.
- Users that were acting unusual in less distinct ways - due to user being mislabeled.
- Users accessing an unusual set of modules - due to testing a new mobile application.

These instances happened to be not malicious. However, an equally plausible explanation could be ascribed to any of them: shared or stolen credentials, patient accessing unlocked workstation, hackers, staff member attempting identity theft. A team tasked with monitoring for inappropriate activity would certainly find this information of value.

Keep in mind that the system produced meaningful results without the need for rule-based reporting. The system analyzed physicians as effectively as it analyzed nurses. The system also did not need any special information on what it meant to be a physician or how a physician was supposed to act.

Additionally the system was able to successfully parse 65,000 instances over 2 months and produced a list of 10 instances that were truly suspicious. This is a massive reduction in work. It would leave security analysts with a manageable amount of work while still meeting a reasonable level of auditing.

A. Application

The techniques describe in this paper can be used to build a system to highly automate the effort of finding suspiciously acting users. Security analysts tasked with detecting inappropriate behavior can use this tool to compliment traditional tasks such reporting on user with the highest number of patients or access to high visibility patients.

B. Enhancements

During the course of this exercise, the system provided many unique views into the users' activity that had never been seen before. In addition to looking for inappropriate activity, this system may be used in other way.

1) It was noted that some doctors were using the system in a way significantly different than their peers. This was discussed with the application support staff and this will be reviewed to see if information can be used for training or efficiencies review of the doctors. There is definite value to the organization to make physicians as efficient as possible.

2) Can this model be applied to clinician performance review? For example classification can be run based on the relative effectiveness of a practitioner4. What can a comparison of their behaviors tell us? Does the number of times a physician access radiology reports have any bearing on the doctor's overall effectiveness? If this system were expanded to include other medical applications, it could consider the type of drugs prescribed, consultations made and nurses who interacted with the patient. How do these values relate to the effectiveness of the physician?

3) Can this model be applied to site performance review? Similar to the clinician discussion above, what would a comparison of hospitals within a system show? This could consider a wide variety of inputs such as hours of mandated training, nurse satisfaction and dollars spent on health IT.

4) Is it possible to tune the dataset over time? For example if certain users are known to act significantly different from their peers (but still in an appropriate fashion), this user should be removed from consideration. Over the course of several iterations, pulling these atypical users from the dataset may make a more accurate mapping.

5) This model could certainly be applied to other non-medical systems. How effective would this be in monitoring e-commerce applications, enterprise authentication systems or any other system?

IX. References

[1] K. Gao, T. Khoshgoftaar, and H. Wang, “An Emperical Investigation of Filter Attribute Selection Techniques for Software Quality Classification”, IEEE IRI 2009, July 2009

[2] P. Chan, R Lippmann, “Machine Learning for Computer Security”, Journal of Machine Learning Research, December 2006

[3] I. Witten, E. Frank, M. Hall, “Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)”, Morgan Kaufmann, January 2011

[4] D. Bachrens, T. Schroeter, S. Harmeling, M. Kowanabe, K. Jansen, K. Muller, “How to Explain Individual Classification Decisions”, Journal of Machine Learning, June 2010

[5] J. Kolter, M. Maloof, “Learning to Detect Malicious Executables in the Wild”, Proceeding of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004

[6] X. Xie, “Ideas and Applications on Support Vector Machine Active Learning”, Unpublished

[7] J. Mingers, "An Empirical Comparison of Pruning Methods for Decision Tree Induction", presented at Machine Learning, 1989, pp.227-243

[8] D. Malerba, F. Esposito, G. Semeraro, "A Further Comparison of Simplification Methods for Decision-Tree Induction", Chapter 35 "Learning from Data: AI and Statistics V, Lecture Notes in Statistics", pp 365-374, Springer-Verlag, Berlin, German

[4] There are numerous outcome measurements that could be used for this.