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

Terrence Ziemniak
DePaul University, tmziemniak@yahoo.com

Follow this and additional works at: https://via.library.depaul.edu/tr

Recommended Citation

This Article is brought to you for free and open access by the Jarvis College of Computing and Digital Media at Digital Commons@DePaul. It has been accepted for inclusion in Technical Reports by an authorized administrator of Digital Commons@DePaul. For more information, please contact digitalservices@depaul.edu.
Use of Machine Learning Classification Techniques to Detect Atypical Behavior in Medical Applications

Terrence Ziemniak
tmziemniak@yahoo.com
CSC 695
Master’s Thesis
June 8, 2010

Advisor: Dr Noriko Tomuro
# Index

Executive Summary .................................................................................................................. 3  
1. Introduction ........................................................................................................................... 4  
2. Background ........................................................................................................................... 5  
3. Impetus .................................................................................................................................. 6  
   3.1 HIPAA.......................................................................................................................... 6  
   3.2 Administrative Simplification ....................................................................................... 6  
   3.3 American Recovery and Reinvestment Act ................................................................. 8  
   3.4 Other Drivers .................................................................................................................. 9  
   3.5 Summary ....................................................................................................................... 10  
4. Challenge .............................................................................................................................. 11  
5. Data .................................................................................................................................... 13  
6. System Components ............................................................................................................ 15  
   6.1 PreProcess.pl ............................................................................................................... 15  
   6.2 WekaDBManagement.pl ............................................................................................ 16  
   6.3 MCInfo.pl .................................................................................................................... 17  
   6.4 Database ....................................................................................................................... 17  
   6.5 Classification System ................................................................................................. 19  
7. Exercises ................................................................................................................................ 21  
   7.1 Types of Investigations ................................................................................................. 21  
   7.2 Node Research .............................................................................................................. 21  
   7.3 Summary of Runs ........................................................................................................... 22  
   7.4 Run 43 ....................................................................................................................... 22  
   7.5 Run 38 ....................................................................................................................... 25  
   7.6 Run 46 ....................................................................................................................... 28  
   7.7 Run 47 ....................................................................................................................... 32  
   7.8 Run 48 ....................................................................................................................... 35  
   7.9 Run 42 ....................................................................................................................... 40  
8. Conclusion ............................................................................................................................... 46  
   8.1 Enhancements .............................................................................................................. 46  
9. Appendix ................................................................................................................................. 48  
   9.1 Supporting Documents ............................................................................................... 48  
   9.2 Independent Study Application ............................................................................... 49
Executive Summary

Build an automatic system for detecting atypical behavior within a health care application. The development will consist of the following steps:

1) Preprocess log files
Take the raw TXT log files generated from a medical application and convert them to a format usable by a classification tool (WEKA\(^1\)).
- Reformat the data so that it adheres to structure required by classifier.
- The application’s log files are event based. Will reconstitute data so that it is properly organized for a given data run. For example, data might be manipulated so that it shows activity per doctor for a 24 hour period – number of patients seen, number of workstations signed into, etc.
- Data will be added to help in classification. For example, information will be added that show the user’s job: physician, nurse, biller, etc.

2) Classify
Run the processed log files through the classification engine.
- Will use WEKA to run through a variety of classification models and determine model and configuration that produces optimal results.
- Will attempt to classify various ‘slices’ of data to see which produce the best results. For example, we will compare the classification of user group nurses to the classification of pharmacists.
- System alerts will be actionable items – in that a nurse that doesn’t act like other nurses should be investigated.

3) Tune
Build process to filter out data that should be ignored.
- Allow for marking of ‘false positive’ or ‘false negative’ alerts. Specifically, if an instance is validated as a misclassification, we will remove it from dataset in order to achieve better classification (once classification is rerun).

4) Report
Produce meaningful reports supporting the investigation as to whether alert is appropriate.
- Generate usable data to support investigation of the misclassified instances.

\(^1\) http://www.cs.waikato.ac.nz/ml/weka/
1. 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 will describe the efforts to design, build, and run a system that will detect atypical behavior in a health care application and see if that behavior is indicative of inappropriate activity. 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 accuracy of such a system in detecting inappropriate activity.
2. 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.”

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 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 policies, auditing and access control, but complement this with other reactive controls. A common safeguard used for this is auditing. This will lead to “reasonable and appropriate” controls to enforce security with the ability to look and see 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:

- 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 appropriate or they may 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.

\footnote{2 Hippocratic Oath}

\footnote{3 This language is directly pulled from the HIPAA security ruleset.}
3. Impetus

3.1 HIPAA

The Health Insurance Portability and Accountability Act (HIPAA) was signed into law by President Clinton in 1996. The law was intended to address some major shortcomings in the current health insurance arena. The legislation specifically “provides for improved access, portability, continuity, and renewability of employment-related group health plans. The act also addresses health insurance fraud and abuse, promotes the use of medical savings accounts, improves access to long-term care insurance coverage, and simplifies and coordinates Medicare benefits.”

The HIPAA legislation consists of five titles. Title one, ‘Health Insurance Reform’, was meant to protect employees who switch jobs by mandating continued coverage to ensure there was no gap in coverage. A gap in coverage would then allow insurance companies to deny coverage of preexisting conditions. Insurance companies were now forced to accept consumers who were covered under a previous plan without consideration of pre-existing conditions.

Title two, ‘Administrative Simplification’, mandated standards by which various insurance functions could be done electronically. By driving the industry to electronic transactions, the law hoped to lessen “administrative burden, lower operating costs, and improve overall data quality”. The most significant of these provisions is under the Administrative Simplification from which the privacy and security standards derive.


3.2 Administrative Simplification

In passing HIPAA, the federal government tasked the Department of Health and Human Services with adopting standards in support of the law. Their first objective was to encourage the use of electronic transactions for various health care functions. To that effect, DHHS adopted standards to drive this initiative. On August 17, 2000 the final rule on ‘Standard for Electronic Transaction and Code Set’ was published (65 FR 50312). In it were references to dozens of detailed rule sets that had been defined by various industry groups. DHHS did not in fact design data structures or protocols, but rather worked with the health insurance industry to choose the best among the competing rule sets.

---

4 http://www.amcp.org/data/jmcp/vol5/num2/niecko.html
5 http://www.cms.gov/HealthInsReformforConsume/02_WhatHIPAADoesandDoesNotDo.asp
6 http://aspe.hhs.gov/admsimp/final/txfinal.pdf
7 http://aspe.hhs.gov/admsimp/final/txfinal.pdf
HIPAA was creating an infrastructure by which health insurance companies and entities they deal with could easily send information back and forth. When compared with paper-based processes, electronic transactions would undoubtedly be quicker and cheaper. Hospitals would now have the ability to check real-time if an insurance card was valid, what the subscriber’s co-pay was, and what benefits were covered under the patient’s plan.

However with the ease by which this information can move, law makers realized that the public would have concerns about the use and misuse of their information. Could a pharmaceutical company use this information to target potential customers? Could psychiatric notes be used in a child custody case? Law makers had to include guidelines for the use and protection of this information. To that end, the Administrative Simplification section of HIPAA required the creation of standards for both data privacy and security. The former ensures that protected information is only used in ways authorized by the law, while the later sets rules by which health care organizations must protect the data.

### 3.2.1 Security Standard

Realizing that health care organizations would be accumulating a mass of data that must be protected from misuse, the Department of Health and Human Services was tasked with creating the Security Standard. In April 2003, HHS published its final ‘Security Standards’ document with the intent to “establish a minimum standard for security of electronic health information.”

The standards lay out a couple of dozen requirements for health care organizations. For example they are required to have a termination policy to ensure that accounts are disabled as soon as possible after an employee leaves the company. Another requirement is that companies must have a disaster recovery plan to ensure timely resumption of services after a disaster.

The security standards also require that companies have security incident procedures. Specifically they are required to implement policies and procedures to “identify and respond to suspected or known security incidents; mitigate, to the extent practical, harmful effects of security incidents that are known to the covered entity; and document security incidents and their outcomes.”

The standard goes on to define a security incidents as follows: “the attempted or successful unauthorized access, use, disclosure, modification, or destruction of information or interference with system operations in an information system.”

---

8. 45 CFR 164 Subpart C “Guiding Principles for Standard Selection”
9. 45 CFR §164.308
10. 45 CFR §164.308
This is a difficult standard to meet because it requires organizations to detect when someone successfully accesses medical data that he shouldn’t. On the surface this appears to be straightforward – boiling down to a simple question – “Is Nurse Smith authorized to look at patient Jones’ information”? In the complex world of medical entities with convoluted relationships as well as poorly defined roles, this is an extremely difficult question to answer. In a large hospital where data is accessed thousands of times by a myriad people, devices and locations each and every day, this would need to be automated. How could a computer tell appropriate versus inappropriate access?

There is no good answer to this question. Luckily for health care organizations, HIPAA was considered a toothless law.\textsuperscript{11} The standard had no requirement for external reporting, and enforcement was only complaint-based. This meant that the data leak had to been traceable back to the health care organization – which is extremely difficult in most cases.

In fact 2010, a full six years after HIPAA Privacy rule set went into effect, was the first time a person was sentenced to prison for violation of this law. There were only a handful of other prosecutions resulting in minimal fines. For the time being, there was little reason for organizations to struggle with this problem.

### 3.3 American Recovery and Reinvestment Act

The American Recovery and Reinvestment Act\textsuperscript{12} (ARRA) was passed in 2009 in response the economic crisis. It had the stated goals of job creation and spurring economic activity. As part of this stimulus package, the bill makes billions of dollars available to drive adoption of health information technology within the industry. Law makers also include language meant to address some of the short comings of the original HIPAA privacy and security rules. These provisions are included in a subtitle of ARRA called HITECH.

#### 3.3.1 Health Information Technology for Economic and Clinical Health Act\textsuperscript{13}

These provisions increased the impetus for health care organizations to look for and address inappropriate activity. This was done in two ways. First the act greatly increased the number of entities that needed to comply with HIPAA. Previously only health care provider, insurance companies and clearing houses had to comply with HIPAA requirements. These entities are collectively known as covered entities. Now companies that utilize protected health information on behalf of covered entities must also adhere to the HIPAA requirements. This means that companies that provide billing services, call center services or

\textsuperscript{12}http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=111_cong_public_laws&docid=f:publ005.pdf
\textsuperscript{13}http://waysandmeans.house.gov/media/pdf/111/hitech.pdf
hundreds of other outsourced services for covered entities are now subject to the provisions of the original HIPAA law. The number of entities that were required to comply with these rules grew exponentially overnight.

The second way that HITECH changed how health care organizations dealt with inappropriate activity is that it put strict requirements on breach notification. In cases where breaches were detected, HIPAA did not obligate organizations to notify affected users. Health care organizations were free to notify the affected individual if they chose but were not strictly required to do so. Under ARRA entities are obligated to make a series of notifications that were “made without unreasonable delay and in no case later than 60 calendar days after the discovery of a breach”.

In addition to mandating notifications in a timely fashion, the law set forth who needs to be notified. Organizations that experienced a data breach must notify not only the affected person but also the Secretary of Health and Human Services for any sized breach. Additionally if the incident included more than 500 individuals, organizations must make “prominent media outlets” aware of the breach.

ARRA now required organizations to deal with breaches in a timely fashion. Delayed responses can put the organization at odds with these laws. ARRA has the teeth that HIPAA lacked. Organizations now had to get serious about protecting health care information.

### 3.4 Other Drivers

HIPAA and ARRA have provisions to ensure that private health information is properly protected. However that is not the only type of consumer data held by health care organizations. For the purposes of managing patients’ records and accounts, a large amount of private data must be kept above and beyond health care information. Each of these data types have their own set of mandates requiring organizations to look for inappropriate access.

#### 3.4.1 State Breach Laws

Nearly all states have laws in place requiring business to notify consumers when their personal information has been misappropriated. Though these vary from state to state, they mostly have the same basic requirements. Business that deal with private consumer data must notify affected individuals where their personal information has been misappropriated. In the state of Illinois, personal information includes the patient’s name in combination with his social security number, driver’s license number or credit card number.

---


Organizations with the ability to detect inappropriate activity and respond quickly to contain the situation might limit their liability in regards to this statute.

### 3.4.2 PCI
The Payment Card Industry Security Standards Council is an association of financial institutions interested in development and adoption of security standards for the protection of data accounts. This group developed a robust set of standards, collectively known as Payment Card Industry Data Security Standards or PCI-DSS\(^\text{16}\), which set forth a proscriptive list of safeguards intended to protect these sorts of financial transactions. All merchants that use credit cards are contractually obligated to comply with these rule sets. Non-compliance can result in fines and higher transactional fees. In the worse cases, the organization may be prohibited from using credit cards.

PCI-DSS requires logging of activity on systems that process cardholder or sensitive authentication data. Additionally these logs must be reviewed regularly looking for the purpose of ‘preventing, detecting, or minimizing the impact of a data compromise’.

### 3.4.3 Class Action Lawsuits
There have been several attempts to file civil\(^\text{17}\) or class action\(^\text{18}\) lawsuits in response to breaches of private information. As shown above, health care organizations are now required to publicize breaches of private data. This will become more common as the number of data breaches will unlikely continue.

To protect themselves from these lawsuits, health care organizations must show that they follow best practices in protection and monitoring of this private information.

### 3.5 Summary
Health care organizations are used to dealing with a plethora of regulations and compliance efforts in regards to accreditation, state and federal requirements and insurance mandates. Protection of health care information is one more issue that will have to be dealt with. The more successful these organizations become in detecting inappropriate activity, the better suited they will be to comply with these requirements.

\(^{16}\)https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml
\(^{17}\)http://privacylaw.proskauer.com/tags/data-breach-litigation/
\(^{18}\)http://www.networkworld.com/community/node/18598
4. 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. Any technology or process that is seen as impeding physician work will be viewed unfavorably by physicians. 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 patients’ data based on their relationship to the physician.

This is extremely difficult to implement effectively.

- 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. These companies will hold data from multiple physicians, labs and hospitals.
- 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.

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 to help machine automatically detect patterns and derive meaning from them. In ML, the task of detecting atypical behavior can be done by building a classification model for typical behavior and identifying the misclassified behavior/items. And there are several classification tools in ML. These tools have been successfully applied to many real world problems, for instance:

- Which emails is spam?
- Is a patient better suited for hard or soft contacts?
- Can species of plant be automatically identified based on attributes?

The following describes a system that will attempt to use classification models to identify which users of this medical system are not acting like their peers. This will then allow anomaly detection to shift from rule-based reporting to a more heuristical model that flags abnormal events by comparing a user’s activity to that of others in the same role.
5. Data

Data for this experiment was pulled from a medical application, McKesson Portal. This system is the primary tool used to view data from dozens of medical systems. 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 system produces log files which record various activities. To maintain HIPAA compliance, the system produces very detailed logs. Below is a listing of events captured in the logs.

- 100_Signum – This event is logged when a user successfully authenticates to the application.
- 100_SignOut – This event is logged when a user is signed out of the application. This can be done by the user or as a result of automatic timeout.
- 100_AutoSuspend – This event is logged when a user’s session is suspended after a set period of inactivity.
- 100_LeaveSuspend – This event is logged when a user’s session is removed from a suspended state.
- 100_ManualSuspend – This event is logged when a user chooses to suspend his session.
- 100_ModuleAccess – This event is logged when a user accesses a new module within the application. Modules are specific views to accomplish certain tasks such as patient lists, station census and transcription.
- 100_ChangePatientContext – This event is logged when a user is granted access to a patient’s information. Once granted access the user will be able to view that patient’s data.
- 300_SigninError – This event is logged when an attempted signin fails.
- 400_ModulePrint – This event is logged when a user uses the print function within the module.

Two events make up the vast majority of logged events. 100_ModuleAccess accounts for 83% of the events and 100_ChangePatientContext accounts for 12%. 100_Signum and 100_Signout make up another 3%. No other event accounts for more than 1% of the events.

The raw logs are written to simple text files and the data is XML-formatted. Below is an example of a typical entry. Some details vary based on the specific event. For example a 100_Signum event does not record a patient’s name.

```
20091001:000221.190:<!xml version="1.0" encoding="UTF-8"?>
```

19 Note that confidential information such as patient name was omitted in this example.
6. System Components

It was necessary to do a series of pre-processing to prepare the data for analysis.

6.1 PreProcess.pl

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 against other users of the system.

- How many times this day did Dr Smith sign in?
- How many patients were viewed by Dr Smith?
- How many workstations did Dr Smith use this day?
- How does this compare to other physicians?
- How does this compare to nurses? Or receptionist? Or 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.

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 for an instance. For example the system records the total number of unique patients accessed by the user over the course of the day. This aggregate information can be significantly different from the simple totals. A nurse in an intensive care unit (ICU) may work with only a dozen patients over the course of the day, but each time she changes context to another patient a 100_ChangePatientContext event is logged. So in a typical day, this ICU nurse 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.

The source for role information came from the organization’s Windows AD system. For each user found in the logs, the user account name was queried in Windows. The user account description field was parsed looking for certain
text strings indicating the role. The text strings were the bases for assigning a value to the data point called RoleAD.

Note that the application does in fact record a RoleID attribute with each log entry. This is intended to be used by the system administrators to document the role information for a given user account. However, in this health care organization’s implementation, this field was not consistently used and was considered less reliable than Windows AD.

The PreProcess script\textsuperscript{20} generates output\textsuperscript{21} written to a comma separated value (CSV) file.

\textbf{6.2 WekaDBManagement.pl}

The classification engine used (WEKA) is able to pull data from a variety of sources. It would be able to pull data directly from the PreProcess data file. However it was desirous to be able to initiate classification runs on various views of the data without having to reprocess the log files.

To achieve this the PreProcess output needed to be imported in a database. Then by the use of SQL calls, the necessary flexibility was achieved. So instead of being restricted to considering the whole data set, it was possible to slice the data in many different ways.

- Look at all sites together, any single site, or a combination thereof.
- Dynamically change the class being classified. That allowed, for example, the analysis of physician’s activity followed by the analysis of pharmacists'.
- Look at different date ranges.
- Provided ability to ignore certain instances or users.

The PreProcess output was imported into the database by means of the WekaDBManagement script. This populated two different tables: instances and users. The instances table contains the relevant information on the various instances. While this table was being populated, the user information was reviewed and added to the users table. In addition to a single entry for each user, summary information over the entire 60 day dataset was collected such as the average number of logins and the average number of patients. This summary information is intended to provide a view into the user’s typical behavior.

Additionally the system flagged cases where there were changes to the user information. For example the system would flag cases where the user signed into workstations at more than one location. These are written to the events table.

\textsuperscript{20} The PreProcess script can be found in the supporting documents.
\textsuperscript{21} The PreProcess output file can be found in the supporting documents.
6.3 MCInfo.pl

For the purposes of easily retrieving information needed for analysis, a script was written which pulled relevant information from the raw log files. The script pulls information from ARFF files (WEKA output) and prompts the user to provide details necessary to narrow down the request.

- **FileName**: Path and filename of ARFF file
- **Analysis Type**: Instance will report on single instance. Classification will report on entire leaf. User will pull up all instances associated with that particular user.

The script will prompt the user for information based on the type of analysis requested. At the end of each request, the raw log files will be presented allowing the analyst to see the detailed activity.

6.4 Database

The database holds all information regarding instances, users and any tuning efforts. It consists of three tables: instances, users, and events.

6.4.1 Instance Table

The instances table holds the individual instances derived from the application logs. The table is populated by the WekaDBManagement script.

```sql
CREATE TABLE `weka`.`instances` (
`User` varchar(20) NOT NULL,
`LogFile` int(11) NOT NULL,
`Logins` int(10) NOT NULL,
`Hours` int(10) NOT NULL,
`LogOuts` int(10) NOT NULL,
`Suspends` int(10) NOT NULL,
`ModulesUnique` int(10) NOT NULL,
`ModulesTotal` int(10) NOT NULL,
`PatientsUnique` int(10) NOT NULL,
`PatientsTotal` int(10) NOT NULL,
`APWW` int(10) NOT NULL,
`PrintModule` int(10) NOT NULL,
`WorkstationsUnique` int(10) NOT NULL,
`IgnoreInstance` int(10) NOT NULL DEFAULT '0')
```

All of these values are pulled directly from the log files and represent totals for that log file – a 24 hour period. Additionally the IgnoreInstance attribute was added to support tuning efforts -- this column is used to exclude specific instances when importing data into WEKA, because it may be of value to ignore certain data while analyzing specific instances.

6.4.2 Users

The users table holds information regarding the individual users of the system. This table is populated by the WekaDBManagement script. Whenever a record is added to the Instances table, the scripts looks to see if a corresponding user
exists. If the a record doesn’t exist it will be added. If a record does exist for this user, then the values will be updated.

```sql
CREATE TABLE `weka`.`users` (
    `User` varchar(20) NOT NULL,
    `InstanceCount` int(11) DEFAULT '1',
    `LoginsAvg` int(11) DEFAULT NULL,
    `HoursAvg` int(11) DEFAULT NULL,
    `LogOutsAvg` int(11) DEFAULT NULL,
    `SuspendsAvg` int(11) DEFAULT NULL,
    `ModulesUniqueAvg` int(11) DEFAULT NULL,
    `ModulesTotalAvg` int(11) DEFAULT NULL,
    `PatientsUniqueAvg` int(11) DEFAULT NULL,
    `PatientsTotalAvg` int(11) DEFAULT NULL,
    `APWWAvg` int(11) DEFAULT NULL,
    `PrintModuleAvg` int(11) DEFAULT NULL,
    `WorkstationUniqueAvg` int(11) DEFAULT NULL,
    `Site` int(11) DEFAULT NULL,
    `RoleAD` int(11) DEFAULT NULL,
    `RoleID` int(11) DEFAULT NULL,
    `Groups` varchar(3000) DEFAULT NULL,
    `IgnoreUser` int(10) DEFAULT '0',
    PRIMARY KEY (`User`))
```

Most of the data in a given record consists of aggregate data for the user across the entire range of logs. This provides a view into the average number of logins (LoginsAvg) or the average number of print jobs (PrintModulesAvg) for the user. This information is intended to show typical behavior for the user and will be useful when analyzing misclassified instances.

Additionally there are a few columns that help identify the user’s role. RoleAD is used to assign the user a role based on the information pulled from the organization’s Windows AD infrastructure\(^\text{22}\). RoleID assigns the user a role based on information maintained in the application’s user table\(^\text{23}\). Groups is a listing of all AD groups which the user account is included in. As the organization uses AD groups to assign access to many resources, this information was captured to see if this might be a more accurate method of ascribing roles to users.

Lastly there is a column that records the site which the user works at. This was determined based on the workstations the user signed into. If the user signed into workstations from multiple sites, a record was added to the event table.

### 6.4.3 Events

In addition to flagging when a user works at more than one site, the WekaDBManagement script looks for other data changes that might indicate unusual activity. Specifically changes to RoleAD, RoleID, and Groups are also

\(^\text{22}\) Mapping of AD to role can be found in FindUserComments subroutine of the PreProcess script.

\(^\text{23}\) Details on the application roles can be found in the RoleID Codes spreadsheet.
logged. This was intended to show another view of the user’s activity that might be useful when analyzing misclassifications.

```sql
CREATE TABLE `weka`.`events` (
  `Type` varchar(45) NOT NULL,
  `Description` varchar(3000) NOT NULL)
```

Lastly it should be noted that no confidential patient data is held in the database. This data is captured in the original application logs but is removed by the Preprocess script. Below is a data architecture diagram showing the various components described above.

### 6.5 Classification System
The WEKA\(^{24}\) 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 ML 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, not only because it is scalable to large datasets, it also produces output (i.e. decision tree) that can be easily understood by a user. It is expected that by analyzing the decision tree, the user can make educated decisions regarding likely candidates (atypical behavior) for further investigation.

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 the instances. 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. The overall accuracy shows how accurate the tree classified all users. Additionally there is a separate accuracy rate for the users in a specific role.

For example if a tree attempts to classify physicians and has an overall accuracy rate of 80%, that 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 was a physician, the system would correctly classify them nine out of ten times\(^{25}\). 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.

\(^{25}\) Conversely a non-physician would be classified correctly with less than 80% accuracy.
7. 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 represent inappropriate activity. The efforts documented below start with a description of each run. This describes the starting data set and the initial classification run. For example a run might attempt to analyze all nurses at site 4. Subsequent to each run is a series of investigations. These are attempts to derive meaning from the results of the classification run. So an investigation might attempt to determine why a certain number of nurses were incorrectly classified.

7.1 Types of Investigations

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

The second method of investigation incorporates consideration for the length of the path. 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 false-positive would be the case where a nurse acts like a physician, and an example false-negative would be the case where a physician does not act like his peers. 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.

7.2 Node Research

Once it is decided that a node is to be investigated, two primary tools are used for research. First the raw log files are pulled either by means of MCInfo.pl or by simple searching through the 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 users. 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.

### 7.3 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 43 investigates pharmacists from site 0. Ad-hoc investigations found two suspicious instances.
- Run 38 classifies residents at site 4. Ad-hoc investigations discovered several users with suspicious activity.
- Run 46 combines residents and physicians at site 4 in a single classification group. The path-length investigations produced four suspicious instances.
- Run 47 considers physicians of site 6. Several path-length investigation uncovered several cases of suspicious activity.
- Run 42 investigates physician assistants from site 0. The filtered-classification investigations did produce one suspicious instance.
- Run 48 investigates case-workers from site 2. This also used the filtered-classification method, and flagged four suspicious instances.

Note that the Weka files and spreadsheets associated with these runs are available in the supporting documents section.

### 7.4 Run 43

This run analyzes pharmacists from site 0. It was selected to examine how the system would treat a smaller user group. For this site pharmacists account for only 202 of the 9620 instances – 2.1%. The following SQL code was used for the date import.

```sql
SELECT logins, hours, logouts, suspends, modulesunique, modulestotal, patientstotal, patientsunique, apww, printmodule, workstationsunique,
```
if(rolead = 8, '1', '0') as class
from instances, users
where instances.user = users.user
and site = 5

Attribute filtering was attempted but was found to significantly reduce the accuracy. As shown above, the entire set of attributes were used.

The J48 classifier (with parameters of confidenceFactor of .05 and minNumObj of 5) was applied and produced an overall accuracy rate of 99.3%. The classification accuracy of actual pharmacists was 79.9%. The following tree was generated.

```
7.4.1 Investigation 1
Many instance followed the path to the node [0(8206/19)]. The system predicted these instances were of role 0 (not pharmacist). Of the 8206 instances, 19 were misclassified resulting in a high accurate rate of 99.7%. Note that there are only 13 pharmacists in the site and they accounted for only 202 of the 9620 instances. The following SQL call was used to display the misclassified users. Notice that the SQL code directly maps out the path to the particular leaf.

select * from instances i, users u
where i.user = u.user
and site = 5
and modulesunique > 7
and rolead = 8
```
The activity for the 19 pharmacists did not show any significant differences when compared to either the users’ typical behavior or to the typical behavior across all pharmacists.

### 7.4.2 Investigation 2

This node [1(130/3)] shows a group of instances that had a high accuracy rate (97.6%) and only three misclassified instances. The following SQL call was used to display the misclassified users.

```sql
select * from instances i, users u
where i.user = u.user
and site = 5
and modulesunique = 7
and patientsunique > 4
and printmodules <= 0
and rolead <> 8
```

The three instances represent two users, one a physician and the other a Quality Outcomes Case Manager. The activity for these three instances was not out of line based on the users’ previous behavior.

### 7.4.3 Investigation 3

This node [1(50/15)] shows a group of instances that had a lower accuracy rate (70%). It was selected for investigation to see whether or not the nodes with lower accuracy rates produce better information than high accuracy nodes. This node had 15 misclassified non-pharmacists. The following SQL call was used to display the misclassified users.

```sql
select * from instances i, users u
where i.user = u.user
and site = 5
and modulesunique = 7
and patientsunique <= 4
and suspends > 0
and logouts <= 2
and printmodule <= 0
and rolead <> 8
```

The following observations can be made regarding this node.
- One user (afeokt01) had numbers that were much lower than his typical work day. It was noted that this user’s shift was much shorter than normal that day.
- One user (krobles) had three separate instances in this node. One shift had one metric, ModulesTotal, much higher than normal even though his shift length was normal. However this is not considered inappropriate
since the user did not have a rise in the number of ModulesUnique (i.e., he was working within his normal set of modules).

7.5 Run 38
This run analyzes residents (physicians in training) from site 4. The following SQL code was used for the date import.

```
SELECT hours, modulesunique, modulestotal, patientstotal, APWW, PrintModule, WorkstationsUnique,
    if(rolead = 2, '1', '0') as class
FROM instances, users
WHERE instances.user = users.user
    and ignoreinstance = 0
    and ignoreuser = 0
    and site = 4
```

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

![Tree Diagram]

The question was then asked: what information can be gleaned from this result set. Furthermore can this be used to detect inappropriate activity? The activity associated with several leafs were investigated.
7.5.1 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 showed 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? The results of the above SQL call allowed us to make the following observations.

- User sjabsh01 had 50 recorded instances over the entire data set. The misclassified instances were consistent with this historical behavior.
- User pkarth01 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 shorted day.

There appears to be no activity of concern in this leaf.

7.5.2 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 (i.e. class 1). 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.
• User ccosta01 had 54 instances over the entire dataset. The user’s activity was consistent with his historical patterns.
• User agupta01 had 51 instances over the entire dataset. The user’s activity was consistent with his historical patterns.
• User mhyser01 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 User’s activity for the day was pulled from the raw logs. The activity which deviated from historical norms for that user was reviewed.
  o The number of unique modules, 27, substantially deviated from his historical activity. Below is a breakdown of the user’s behavior over the entire dataset.

<table>
<thead>
<tr>
<th>ModulesUnique</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
</tr>
</tbody>
</table>

o However the number of patients did not deviate. Upon review of the unusual modules seen by this user, there is no evidence of suspicious activity. Below is a listing of the modules viewed by this user.

<table>
<thead>
<tr>
<th>Typical</th>
<th>Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Announcement - GE Muse</td>
</tr>
<tr>
<td>N</td>
<td>Announcement - Medication Orders</td>
</tr>
<tr>
<td>Y</td>
<td>Announcement - Patient Search</td>
</tr>
<tr>
<td>N</td>
<td>Announcement - Physician Search</td>
</tr>
<tr>
<td>Y</td>
<td>Announcement - Physicians</td>
</tr>
<tr>
<td>Y</td>
<td>Announcement - Results</td>
</tr>
<tr>
<td>N</td>
<td>Announcement - Transcriptions Alert</td>
</tr>
<tr>
<td>Y</td>
<td>CM MPI/Active Search</td>
</tr>
<tr>
<td>N</td>
<td>CM Patient Demographics</td>
</tr>
<tr>
<td>N</td>
<td>CM Staff Directory</td>
</tr>
<tr>
<td>Y</td>
<td>CM Station Census</td>
</tr>
<tr>
<td>N</td>
<td>CM Summary Viewer</td>
</tr>
<tr>
<td>Y</td>
<td>CM Working Patient List</td>
</tr>
<tr>
<td>N</td>
<td>GE Muse Web Results</td>
</tr>
<tr>
<td>Y</td>
<td>Login</td>
</tr>
<tr>
<td>Y</td>
<td>My Personal Notes</td>
</tr>
<tr>
<td>Y</td>
<td>My Web Page Links</td>
</tr>
<tr>
<td>N</td>
<td>OLR/RMC/SFH Clinical Documentation</td>
</tr>
<tr>
<td>Y</td>
<td>Patient Demographics with Images</td>
</tr>
</tbody>
</table>
There is a node \([0 (253/82)]\) where 82 residents were misclassified. The system predicted that these users were not residents (i.e. class 0). The classification had a relatively low accuracy rate of 36 out of 38 times (67.5%).

The following observations were made regarding this node.
- One user (mpongr01) 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.

### 7.6 Run 46

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 grouped based on a single attribute? What does it mean when a group of instances require nine decision points to be classified? The following two runs will compare nodes at different path lengths (where a path length is the length of the path from the root of the tree to a node) to see if the number of decision points would help improve the usefulness of the tool for finding inappropriate behavior.

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

```sql
SELECT modulesunique, patientstotal, APWW, printmodule,
if(rolead = 2, '1', if(rolead = 1, '1', '0')) as class
from instances, users
where instances.user = users.user
```
This has a small number of attributes. The original list of 14 attributes\(^{26}\) was filtered down to these three by means of WEKA’s attribute filter function and parameters. This filtering is the system’s guess at which attributes are most relevant for classification purposes.

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.

7.6.1 Investigation 1

The first investigation deals with nodes with short paths. The node [1(4191/352)] classified role 1, physicians and residents, with an accuracy of 91.7%. The classification was based on a single criterion: PatientsTotal > 7.

```sql
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.

\(^{26}\) 14 is the number of columns in the Instances table.
User bstrock 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.

User avadineanu 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 works for medical records and has similar work days about every 2 weeks. This was considered normal activity for this user.

7.6.2 Investigation 2

This investigation deals with nodes with medium length paths. The first node \( [0(19/2)] \) classifies role 0, not physicians or residents, with an accuracy of 89.5%. The classification is based on four decisions: PatientsTotal \( \leq 7 \), ModulesUnique \( > 24 \).

```sql
SELECT * from instances i, users u
where i.user = u.user
and site = 4
and patientstotal \( \leq 7 \)
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 role 1 (i.e. physicians or residents) with an accuracy of 99.4%. The classification is based on five decisions: PatientsTotal \( \leq 7 \), ModulesUnique \( \leq 10 \), PatientsTotal \( > 1 \), ModulesUnique \( > 5 \), ModulesUnique \( > 4 \).

```sql
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 (jjohnson) 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 (RoleAD) were blank. It is likely that this user was correctly classified by WEKA but simply mislabeled in RoleAD.

### 7.6.3 Investigation 3

This investigation deals with nodes with longer length paths. The first node [1(127/41)]) classifies role 1, physicians and residents, with an accuracy of 67.8%. The classification is based on eight decisions: PatientsTotal <= 7, ModulesUnique <= 10, ModulesUnique > 5, PatientsTotal > 3, ModulesUnique > 8, ModulesUnique > 9, PrintModule <= 2.

```sql
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%. The classification is based on seven decisions: PatientsTotal <= 7, ModulesUnique <= 10, PatientsTotal <= 1, PatientsTotal <= 0, ModulesUnique > 2, ModulesUnique > 5, ModulesUnique > 6.

```sql
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 unexpected 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.
7.7 Run 47

Run 47 is the second case where nodes at different depths of the decision tree are compared. This run analyzes physicians from site 6. The following SQL code was used to populate WEKA for this run.

```
SELECT modulesunique, patientsunique, APWW, PrintModule,
       if(rolead = 1, '1', '0') as class
from instances, users
where instances.user = users.user
and site = 6
```

This has a small number of attributes. Filtering was done by manual inspection and testing to find a group of attributes that produced a high level of accuracy while creating a relatively small decision tree.

The J48 classifier (with parameters of confidenceFactor of .05 and minNumObj of 5) produced an overall accuracy of 74.9% and the true positive classification of physicians with an accuracy of 95.0%. The following tree was produced.

```
    +--- role ad
    |     +--- APWW <= 0
    |         +--- PrintModule > 7
    |               +--- ModulesUnique > 15
    |                   +--- rolead = 1
    |                       +--- APWW <= 10
    |                           +--- PrintModule <= 7
    |                               +--- ModulesUnique <= 15
    |                                   +--- rolead = 1
    |                                       +--- APWW <= 10
    |                                           +--- PrintModule <= 7
    |                                               +--- ModulesUnique <= 15
    |                                                  +--- rolead = 1
    |                                                      +--- APWW <= 10
    |                                                          +--- PrintModule <= 7
    |                                                              +--- ModulesUnique <= 15
    |                                                                  +--- rolead = 1
    |                                                                      +--- APWW <= 10
    |                                                                            +--- PrintModule <= 7
    |                                                                                   +--- ModulesUnique <= 15
    |                                                                                     +--- rolead = 1
    |                                                                                         +--- APWW <= 10
    |                                                                                               +--- PrintModule <= 7
    |                                                                                                               +--- ModulesUnique <= 15
```

7.7.1 Investigation 1

The first investigation deals with nodes with short paths. The node [1(22/3)] classifies role 1, physicians, with an accuracy of 85.7%. The classification is based on three criteria: APWW <= 0, PrintModule > 7 and ModulesUnique > 15.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 6
and rolead <> 1
and apww <= 0
and printmodule > 7
```
Of these 3 misclassified instances these observations were made.

- Two users (ajoba01, adaggu01) had higher than normal counts for PatientsUnique and PatientsTotal.
- These two users had higher than normal counts for ModulesTotal. Their numbers for ModulesUnique were normal.
- The hours worked for these two were higher than normal, which accounts for the above mentioned behavior.
- There third user (bboncz01) had no substantial change to his typical behavior.
- These three users are all physician residents. It is not unexpected that their behavior would be similar to that of physicians.

The second investigation was done to review with nodes with short paths. The node \[0(140/11)] classifies role 0, non-physicians, with an accuracy of 92.1\%. The classification is based on three criteria: APWW \(\leq\) 0, PrintModule > 7 and ModulesUnique \(\leq\) 15.

Of these 11 misclassified instances only a single point stood out.

- A user (mmoren01) had a large increase in his usual number of print jobs. Upon review this physician had one day per month with a large number of print jobs. This is likely due to his printing as a means of billing.

### 7.7.2 Investigation 2

This investigation deals with nodes with medium length paths. The first node \[0(104/10)] classifies role 0, non-physicians, with an accuracy of 90.3\%. The classification is based on six decisions: APWW \(\leq\) 0, PrintModule \(\leq\) 7, ModulesUnique \(\leq\) 9, PatientsUnique \(\leq\) 7, ModulesUnique \(\leq\) 5, ModulesUnique \(\leq\) 4.
and printmodule <= 7
and modulesunique <= 4
and patientsunique <= 1

Of the 10 misclassified instances these observations were made.
- All 10 instances show physicians in a shorter than normal work day.
- Some of these instances show physicians working with the system but not registering a single 100_ChangePatientContext event. This is rare but not unexpected.

The second investigation deals with nodes with medium length paths. The first node \( [1(323/43)] \) classifies role 1, physicians, with an accuracy of 86.6%. The classification is based on six decisions: APWW <= 0, PrintModule <= 7, ModulesUnique <=9, PatientsUnique <= 1, ModulesUnique <= 5, ModulesUnique > 4.

SELECT * from instances i, users u
where i.user = u.user
and site = 6
and rolead <> 1
and apww <= 0
and printmodule <= 7
and modulesunique =5
and patientsunique <= 1

Of the 43 misclassified instances these observations were made.
- In general the decisions are limiting by nature. That is the decision path tends towards small values (e.g. APWW >= 0, PatientsUnique <=1). Nothing in the misclassified instances indicates any inappropriate activity.
- On average the total hours for the individuals is almost 1.5 hours less than each user’s average.
- 28 instances are either residents (physicians in training) or are actual physicians that were mislabeled.

7.7.3 Investigation 3
This investigation deals with nodes with longer length paths. The node \( [1(85/15)] \) classifies role 1, physicians, with an accuracy of 82.3%. The classification is based on eight decisions: APWW <= 0, PrintModule <= 7, ModulesUnique > 9, ModulesUnique > 10, ModulesUnique > 13, ModulesUnique <=22, PrintModule > 2 and PatientsUnique > 8.

SELECT * from instances i, users u
where i.user = u.user
and site = 6
and rolead <> 1
and apww <= 0
and printmodule in (3, 4, 5, 6, 7)
and modulesunique > 13
and modulesunique <= 22
and patientsunique > 8

Of these 15 misclassified instances these observations were made.
- All 15 of these instances were due to residents. It is not unusual that these users act like physicians.
- The individual instances are in line with historical average for each user. No inappropriate activity was found.

The second investigation deals with nodes with longer length paths. The first node \([0(75/29)])\) classifies role 0, non-physicians, with an accuracy of 82.3%. The classification is based on eight decisions: APWW <= 0, PrintModule <= 7, ModulesUnique > 9, ModulesUnique > 10, ModulesUnique > 13, ModulesUnique <=22, PrintModule > 2 and PatientsUnique <= 8.

SELECT * from instances i, users u
where i.user = u.user
and site = 6
and rolead = 1
and apww <= 0
and printmodule in (3, 4, 5, 6, 7)
and modulesunique > 13
and modulesunique <= 22
and patientsunique <= 8

Of these 29 misclassified instances these observations were made.
- None of these instances are out of line with the user’s historical behavior. These all appear to be non-malicious.

7.8 Run 48

The following two runs took a different approach. It was asked what filtering can be done to make the model more accurate. Can instances that are statistical outliers be ignored? How much more effective would a decision tree be if it ignored those outliers?

In the following two examples, two decision trees were built. The first follows earlier examples. Then another tree was built using only the correctly classified instances in the first tree. Detailed investigations were then run on the resulting decision tree.

This run analyzes case workers from site 2. For this site case workers staff
account for 379 of the 9670 instances (3.9%). The following SQL code was used for the date import.

```sql
SELECT logins, suspends, modulesunique, modulestotal, patientsunique, patientstotal, APWW, PrintModule, WorkstationsUnique, if(rolead = 13, '1', '0') as class
from instances, users
where instances.user = users.user
and site = 2
```

Attribute filtering was attempted, but was found to significantly reduce the accuracy. The set of attributes used were found by trial and error to produce the highest accuracy while creating the smallest tree.

The J48 classifier (with parameters of confidenceFactor of .05 and minNumObj of 5) was applied and produced an overall accuracy rate of 96.8%. The classification accuracy of actual medical records staff was 47.0%. The following tree was generated.

To pull out instances that were misclassified, it was necessary to modify the ARFF file that was produced by the initial tuning exercise. All instances that were misclassified in this run were then removed (303). This included both case workers misclassified as non-case workers as well as non-case workers misclassified as case workers. The data set was then reclassified by WEKA.

The J48 classifier (with parameters of confidenceFactor of .05 and minNumObj of 5) was applied and produced an overall accuracy rate of 99.6%. The
classification accuracy of case workers was 88.7%. The following tree was generated.

### 7.8.1 Investigation 1

Following the previous example, two nodes were examined at the top, middle, and bottom of the tree, respectively. The first investigation for a short path was node [0(8561/1)]. This node contained a single case worker that was misclassified.

```
SELECT * from instances i, users u
where i.user = u.user
and site = 2
and rolead = 13
and ignoreinstance = 0
and apww <= 0
and patientsunique > 0
```

The following observations were made regarding this node.
- The user (tdavis09) had higher than usual counts for most attributes. But the user also worked a day that was double a typical shift. The numbers for this instance are reasonable considering the extended shift.

### 7.8.2 Investigation 2

This investigation examined paths of medium length. The first node [0(11/2)] for consideration contained two case workers that were misclassified.
The following observations were made regarding this node.

- The two users are in fact the same user (tdavis09) discussed above. In these two cases the user had a much shorter shift than normal so the counts appear appropriate.
- It should be noted that this user has a total of 24 instances in the data set. Of these, 21 were removed as part of the initial tuning. The remaining 3 were discussed in the previous two investigations.

The second node for consideration [0(156/8)] contained 8 case workers that were misclassified.

The following observations were made about this node.

- Most of these users had noticeable changes in their typical behavior, but additionally had worked an atypical shift – either longer or shorter than normal. There was no inappropriate activity found.
- It was found that these users were tagged as case workers in the applications (i.e. RoleID) but were tagged as either nurses or medical records staff in Windows (i.e. RoleAD).

The third node for consideration [0(32/3)] contained 3 case workers that were misclassified.
and rolead = 13
and ignoreinstance = 0
and apww > 0
and patientsunique <= 20
and suspends > 0
and workstationsunique > 1

The following observations were made about this node.

- Similar to the previous nodes, there was no sign of inappropriate activity in these instances.

**7.8.3 Investigation 3**

This investigation examined paths of longer length. The first node [1(157/17)] for consideration contained 17 non-case workers that were misclassified.

SELECT * from instances i, users u
where i.user = u.user
and site = 2
and rolead <> 13
and ignoreinstance = 0
and apww > 0
and patientsunique <= 17
and suspends > 0
and workstationsunique <= 1

The following observations were made regarding this node.

- Most of these users had noticeable changes in their typical behavior, but additionally had worked an atypical shift – either longer or shorter than normal. There was no inappropriate activity found.
- There were three users that accounted for 12 of the instances.

The final node for consideration [1(6/1] contained a single non-case worker that was misclassified.

SELECT * from instances i, users u
where i.user = u.user
and site = 2
and rolead <> 13
and ignoreinstance = 0
and apww > 0
and patientsunique in (18, 19, 20)
and suspends > 0
and workstationsunique <= 1
and modulesunique > 14
The following observations were made about this node.

- The user (lstopka) had higher than average counts for most attributes.
- It is interesting to note that this user (lstopka) accounted for five of the 17 misclassified instances in the first run of investigation 3 (immediately prior).

### 7.9 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). For this site physician assistants accounted for only 7882 of the 19295 instances. 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
```

Attribute filtering was applied and resulted in the select statement shown above. Also this run did not use the typical J48 parameters. The original runs produced very large trees 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.
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.
7.9.1 Investigation 1

Following the previous example, two nodes were examined at the top, middle, and bottom of the tree, respectively. The first investigation for a short path was node [1(162/11)]. This node contained 11 non-physician assistants that were misclassified.

```
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 (lnuth01). 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.

```sql
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 (cgo01) 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.

### 7.9.2 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.

```sql
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 (mhambl01) accounted for 4 of these instances. In two 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 previously discussed user (cgo01) has another instance in this node. During the overall tuning, 7 of her instances were set to ignore – leaving 42 instances for this second pass.

### 7.9.3 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.

```sql
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 (mthoma03) with a job description of data assistant. This user’s activity is consistent with his historical behavior, aside from the fact that he printed 4 times when he averages 0. The 4 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.

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.

### 7.9.4 Outliers

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 (bblack01) 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 3 for the other days.
- User (lbialka) 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.
8. Conclusion

This paper described the design and use of a system that could 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.

Once the classification runs identified atypical instances, investigations were called out to those instances which were suspicious. All three investigation types produced instances that were suspicious. The ad-hoc method was the most fruitful. This was likely due to the analyst’s intuitive review of the decision tree. Conversely the filtered-classification was the least fruitful.

Of the roughly two dozen suspicious instances, none were in fact inappropriate in our experiment. Here are some examples of flagged instances that were suspicious but not inappropriate.

- A physician printing out dozens of reports for billing purposes.
- A user testing a new set of modules.
- A user whose role was mislabeled in the application.
- Users with very short or very long shifts
- Many cases where a user's legitimate activity was truly erratic and atypical when compared to his peers.

So the system successfully detected atypical behavior, and the investigations highlighted those atypical instances that were suspicious, but none of these turned out to be inappropriate. Determination of appropriateness requires intense domain and context knowledge. What were the events of day that led a nurse to look at a given record? This is a question that cannot be answered by an automated system.

Overall the system did successful parse 65,000 instances over 2 months and produced a list of 20 instances that were suspicious. This is a massive reduction in work that would leave security analysts with a manageable amount of work while still meeting a reasonable level of auditing.

8.1 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 various other ways, for example:

1. It was noted that some doctors were using the system in 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 comparing effective doctors and ineffective doctors. 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 the system were expanded to include other systems, 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 all sorts 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 safe but atypical users from the dataset may make a more accurate mapping.

5. This system was able to detect when someone’s activity changed overtime. What does it mean if the user’s average print jobs changes significantly? It would be nice to analyze decision trees to see how a suspect user compares to his peers.

6. Would combined filtering produce better results? It was shown that filtered-classification produced some meaningful results but missed some suspicious outliers. Running ad-hoc analysis followed by filtered-classification would likely produce better results.

---

27 There are numerous outcome measurements that could be used for this.
9. Appendix

9.1 Supporting Documents

<table>
<thead>
<tr>
<th>Document Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PreProcess.pl</td>
<td>PreProcess PERL script</td>
</tr>
<tr>
<td>PreProcessOut.csv</td>
<td>Output of PreProcess activities</td>
</tr>
<tr>
<td>WekaDBManagement.pl</td>
<td>WekaDBManagement PERL script</td>
</tr>
<tr>
<td>RoleID Codes.XLS</td>
<td>Listing of application IDs</td>
</tr>
<tr>
<td>43.ARFF</td>
<td>Output of WEKA for run 43</td>
</tr>
<tr>
<td>38.ARFF</td>
<td>Output of WEKA for run 38</td>
</tr>
<tr>
<td>46.ARFF</td>
<td>Output of WEKA for run 46</td>
</tr>
<tr>
<td>47.ARFF</td>
<td>Output of WEKA for run 47</td>
</tr>
<tr>
<td>42.ARFF</td>
<td>Output of WEKA for run 42</td>
</tr>
<tr>
<td>42-Tuned.ARFF</td>
<td>Output of WEKA for run 42 (after tuning)</td>
</tr>
<tr>
<td>48.ARFF</td>
<td>Output of WEKA for run 48</td>
</tr>
<tr>
<td>48-Tuned.ARFF</td>
<td>Output of WEKA for run 48 (after tuning)</td>
</tr>
</tbody>
</table>
9.2 Independent Study Application

Summary

Using various information classification methods, architect and prototype system that will automatically classify health care workers and then flag a worker when his activity is atypical to others in a similar role.

Problem Statement

The Health Insurance Portability and Accountability Act (HIPAA) requires health care entities to look for and deal with security incidents. Unfortunately due to the complex nature of health care it is often difficult as well as impractical to enforce strict access controls to patient data. Restricting a physician’s access to medical data could directly impact critical, time sensitive treatment. But this limited control can be abused by health care workers to ‘snoop’ or otherwise access private medical information inappropriately.

This system will be able to detect when a member of a group (such as physicians, nurses, radiologists) acts different then his peers. The intent is to provide health care providers a means of flagging atypical activity to satisfy the HIPAA requirements.

Deliverable

The deliverables will include:

- System as described above with documentation
- Documentation of practical management of such a system including tuning of alerts, time span for optimal analysis, and the management of ‘false positives’ (i.e. incorrectly labeling activity as a negative event).
- Report on effectiveness of the system based on multiple factors such as worker’s attributes (role, training, etc), organizational attributes (comparing different facilities, maturity of role definition within organization, etc) and various classification models

Evaluation

Evaluation of final product will be based on the following:

- Capability to detect atypical activity within the system
- Ability to tune system regarding false positives. Specifically ability to filter out events that are incorrectly flagged as atypical.
• Comprehensive activity report allowing investigator to determine if atypical behavior is inappropriate. That is, does this constitute a HIPAA violation?
• Proposed model can be used with other medical applications