ML-based Cyber Incident Detection for Electronic Medical Record (EMR) Systems

McGlade, D., & Scott-Hayward, S. (2018). ML-based Cyber Incident Detection for Electronic Medical Record (EMR) Systems. *Smart Health*. Advance online publication. https://doi.org/10.1016/j.smhl.2018.05.001

Published in:
Smart Health

Document Version:
Peer reviewed version

ML-based Cyber Incident Detection for Electronic Medical Record (EMR) Systems

David McGlade, Sandra Scott-Hayward*

Centre for Secure Information Technologies (CSIT), Queen's University Belfast, Belfast, BT3 9DT, N. Ireland

Abstract

An upward trend in cyber incidents across both U.K. and U.S. hospitals has been observed since 2015. Attacks range from identity theft to insurance fraud and extortion/blackmail. The Electronic Medical Record (EMR) systems used in hospitals are targeted due to the sensitivity of data within a healthcare setting. This work is motivated by the necessity to protect patient information and to ensure the availability of such EMR systems. A failure in either case can have grave implications for patients being treated and practitioners using the system. In this research, we propose the application of Machine Learning (ML) and Time Series (TS) anomaly detection to the problem of confidentiality and availability attacks on EMR systems. The results presented in this paper indicate that confidentiality incident detection is fully achievable using ML, with Support Vector Machines obtaining the highest accuracy, precision and recall of a number of models tested. Results from the availability prototype show that the detection of a message surge is possible within 10 seconds, by using an Exponential Moving Average implementation to identify anomalies in message flow. This finding paves the way for an automated surge defence to be developed, presenting a significant advance over the manual method used today. The feasibility and practicality of implementing these detection systems in a clinical setting are also discussed with consideration of parameter tuning, skill-sets, and data protection.

*Corresponding author
Email address: s.scott-hayward@qub.ac.uk (Sandra Scott-Hayward)

Keywords: Electronic Medical Record (EMR) Systems, Cyber-incident Detection, Machine Learning, Anomaly Detection

1. Introduction

An upward trend in cyber incidents against U.S. hospitals has been reported [1], with a 63% increase between 2015 and 2016, and an estimated 12 million patient records being stolen. U.K. hospitals have also reported the number of cyber incidents growing from 16 in 2015, to 55 in 2016 [2]. The types of incidents fall into the following broad categories:

- Identity Theft: Medical data is extremely lucrative. It is estimated that health credentials sell online for 10 to 20 times the value of credit card information [3]. An attacker can use a patient's name, personally identifiable information or protected health information to obtain medical services, prescription drugs or healthcare items [4].

- Insurance Fraud: In the U.S., medical information is used to generate false identities for fraudulent billing purposes [4]. Billing and insurance records are often stolen in combination with Electronic Medical Record (EMR) data. This problem is particularly prevalent in China [5].

- Malicious attack: An EMR system may become compromised by an attacker's actions, which could lead to a situation where the victim's health data is in a corrupted state. This can lead to serious consequences such as the victim receiving an incorrect dosage or medication to which they are allergic [6].

- Extortion/Blackmail: Criminals extort money from individuals or healthcare organisations to prevent the release of private medical information or they use the information stolen to directly blackmail individuals such as celebrities or politicians who would not wish their private medical information to be made public [7].

In addition to these attack types, hospital systems have also faced external Denial of Service (DoS) attacks [8], which aim to cause major disruption to the hospital's operation. Furthermore, hospital systems are also susceptible to incidents from trusted insiders [9] or trusted systems within a hospital network, as identified in Section 4.4. Due to the sensitivity of the medical system and the potential damage from these attacks and incidents, detection measures to enable protection of cyber-physical medical systems must be developed. The EMR system threat model is defined in Section 1. This research focuses on detection measures against these types of incidents.

The unique aspect of this research is the demonstration of a practical, rather than a theoretical, application of incident detection against real EMR systems. The following research contributions are presented in this work:

- Data within a Healthcare setting is extremely sensitive and it is infeasible to use real patient data for prototype purposes. For this reason, a synthetic data set has been carefully generated to reflect the real EMR data attributes. This data set is used in the analysis and is made publicly available for future research studies [10]. In addition, recommendations are provided for extension of the data set.

- Machine Learning (ML) is an often hyped technology. This study presents results to separate the hype from reality, whilst highlighting the wider implications, which need to be considered if deploying a ML solution into a clinical setting. A comparison of ML Classification algorithms is provided along with analysis of the findings. ML is used to improve security with regards to protecting patient confidentiality.

- Additionally, Time-Series (TS) Anomaly detection algorithms are examined to identify how they can be applied to mitigate against DoS attacks from Trusted Systems. This study highlights how an Exponential Moving Average algorithm can be used to detect a DoS attack within 10 seconds. TS is used to improve the security of EMR systems against availability incidents.

- The problem definition of 'cyber incident detection in EMR Systems' is further refined to address the question of how feasible, practical and performant detection actually is against an EMR system? Feasibility covers the tooling and data, Practicality focuses on the difficulty and repeatability of applying ML and TS to the problem space, and performance uses measures such as accuracy, precision and recall to determine how well the prototypes perform.

- Successful detection of both confidentiality and availability incidents are presented. Confidentiality incident detection is fully achievable using ML, with Support Vector Machines obtaining the highest accuracy, precision and recall across the models tested. Results from the availability prototype show that the detection of a message surge is possible within 10 seconds, by using an Exponential Moving Average implementation to identify anomalies in message flow.

This paper is organized as follows: Section 2 introduces the related work. The background to EMR systems and relevant ML techniques is introduced in Section 3. In Section 4, the problem is fully defined. The methodology is described in Section 5. The test environment is detailed in Section 6 and results and analysis are presented in Sections 7 and 8. The feasibility of ML-based cyber incident detection is discussed in Section 9. Section 10 concludes the paper.

2. Related Work

Although there has been some study of security with respect to EMR systems, there are a number of gaps. We identify these through an analysis of the current literature related to EMR system security measures and existing EMR system cyber attack detection mechanisms.

2.1. Current EMR system security measures

When reviewing common current security measures that are frequently deployed to protect EMR systems [11, 12, 13, 14], four main themes are prevalent:

Network Security. Open Standards Institute (OSI) Network layer defence mechanisms such as Network Intrusion Detection systems are well understood, and network level attacks are also well known. However, the ability to detect a cyber incident within an EMR system at the OSI Application layer is immature.

Role Based Access Control (RBAC). RBAC is a common security mechanism implemented in EMR systems which provides measures to restrict who can access the system and what resources they can view. A large number of people access an EMR during its lifetime [15], with each action they perform against the EMR generating numerous system audit events. Any form of incident detection mechanism will need to be able to discern malicious behaviour from normal operation, irrespective of the volume of audit records. Further compounding the difficulty of this task is the fact that malicious behaviour could be conducted by a known and trusted user. These conditions are suggestive of the application of Machine Learning (ML) for incident detection; ML can be applied to discern patterns in data that are not easily interpretable by humans, across large volumes of data.

Logging and Monitoring. Logging and monitoring is key for cyber incident detection. It is needed, in particular, to aid in any follow-on forensic analysis in light of a cyber incident, and to establish non-repudiation of user actions.

Encryption. EMR systems typically use strong encryption mechanisms to protect patient data, both in message transmission and storage within the EMR system.

These examples provide reasonable security controls for an EMR system against typical threats, but they do not protect against an adversary who wishes to impact an EMR system by compromising a trusted messaging system already

connected to the EMR system, or from a trusted insider who already has appropriate RBAC permissions to use the EMR system. The threat of masqueraders, traitors, and unintentional perpertrators has been recently analysed in [16].

For the EMR context, a Patient Administration System could be directed to flood an EMR system with a high volume of HL7 messages, causing a DoS attack. Additionally, a practitioner, who has the appropriate level of RBAC controls to access the EMR system, could use their privileges to access the records of patients whom they do not directly care for. These threats are further explained in Section 4.

2.2. Current EMR system cyber incident detection mechanisms

There is no specific literature focused on the detection of incidents stemming from 'trusted systems or trusted users'. However, research from other areas of industry and healthcare can offer insights to aid this study. This is summarized as follows:

Monitor the patient's treatment journey. - Li et al. [17] propose an implementation of a context aware system, which identifies any deviations from a patient's normal treatment journey or workflow as anomalous (e.g. patient admission to treatment, then discharge or transfer). This proposition is supported by [18], which states that "approximately 80% of patients are on such workflows". Although this figure was gained from a 4-month study of a single large hospital, no research was presented to confirm whether other hospitals follow the same workflow journeys. Additionally, patient caregivers and treatments change frequently, which would void any detection model that has been previously generated. It is difficult to see how this approach could be applied to multiple hospitals or be generic enough to be built within an EMR product.

Anomaly detection across multiple input streams. - Mohan [19] proposes using anomaly detection indicators from multiple input streams to determine if malicious intent is occurring. The input streams are:

1. The illness the patient is currently being treated for, compared to the previous illnesses they have presented during past encounters.
2. The insurance details used by the patient within the healthcare provider.
3. The location from where the EMR was updated, compared to previous locations from where the EMR was updated.

This approach shows promise, but would be quite an extensive undertaking due to the integration required between the many healthcare providers and insurance companies involved. Additionally, this approach really only targets external scenarios, i.e. patients committing fraud. It does not cover malicious insiders. Furthermore, clinical knowledge would be required to understand if any illness presented by the patient constitutes an anomaly compared to previous illnesses; a subject area which is non-trivial to analyse, and dangerous to misdiagnose.

Other related works include an enhanced Security Incident Event Management (SIEM) system [20], which uses state-machine mechanisms to detect cyber incidents, and a Community Anomaly Detection System [21], which uses Unsupervised Machine Learning to detect insider threats based on usage patterns within a collaborative working environment.

It is worth noting that a SIEM is a common approach in industry for detecting incidents in IT based systems and a SIEM could feasibly be deployed to look for cyber incidents in an EMR system. However, a limitation on SIEM products is that they are rule-based in nature, which requires human input to create and maintain detection rules. Rules are static, and they may not detect incident patterns that are unknown, too complicated for human operators to perceive, or are carried out by trusted users [22]. This limitation in SIEM technology leads to our further consideration of ML approaches.

The Community Anomaly Detection System [21] advocates using ML to look at cluster groupings between clinicians and patients, and to identify any outliers in group collaboration. An interesting finding from the paper shows that anomaly detection using ML is difficult if a malicious user accesses only a small set of EMR data infrequently. A further finding from this work is that

access to realistic training and testing data was a key challenge. The same issue has been encountered in this study (see Section 4).

Finally, it should be noted that no direct research was found that applied ML directly to cyber incident detection for an EMR system. Any research reviewed was focused solely on network layer anomaly detection, not application layer anomaly detection. However, prior work has been reviewed to identify the potential ML techniques suitable for use in this scenario. These techniques are introduced in Section 3.

3. Background

3.1. What is an EMR?

An electronic medical record (EMR) is a digital version of a patient's paper-based medical record. All patient records are collectively held together within an EMR system. Traditionally, the term EMR can be used to refer to both the medical record, or the system directly.

The EMR system stores a wide range of data on the patient, such as any interactions between a patient and a hospital, known as encounters [23], or any measurements or assertions made against the patient during their visit, that are known as observations [4]. Typically, patients go through a process of being admitted to the hospital, treated for their symptoms and then discharged from the hospital or transferred to an alternative facility. The EMR system is used by a number of people within the hospital (also known as practitioners), such as nurses, clinicians or clerical staff, to record or update information on the patient throughout the patient's visit or visits to the hospital. Typically, a patient's medical record is accessed by up to 400 people throughout their healthcare journey [15]. Evolve EMR [24] and Evolve Integrated Care systems [25] from Kainos Software are used as reference EMR systems in this research. The functions and workflow of the Evolve systems explored in this research are representative of all EMR systems such that the results presented are widely applicable.

3.2. Kainos Evolve EMR and Evolve IC

Evolve EMR [24] is principally used for digitising legacy records, replacing inefficient paper-based processes and enabling access to information at the point of care via electronic forms and workflow. At the time of writing, 110 hospitals are now live with Evolve EMR, storing 33 million patient records and over 1.3 billion documents. Evolve EMR integrates to other hospital systems, such as Patient Administration Systems (PAS), Pathology, Laboratory and Financial systems through the use of the HL7 V2 [26] messaging standard, as shown in Figure 1. HL7 messages are gathered and transformed within an integration engine before being passed onwards into Evolve EMR. Each individual HL7 message received from the Integration Engine causes an action to occur within Evolve. For example, an A01 message will create a patient record within the EMR system and admit the patient, or an A03 message will initiate a workflow within Evolve to generate documentation to transfer a patient to another hospital.

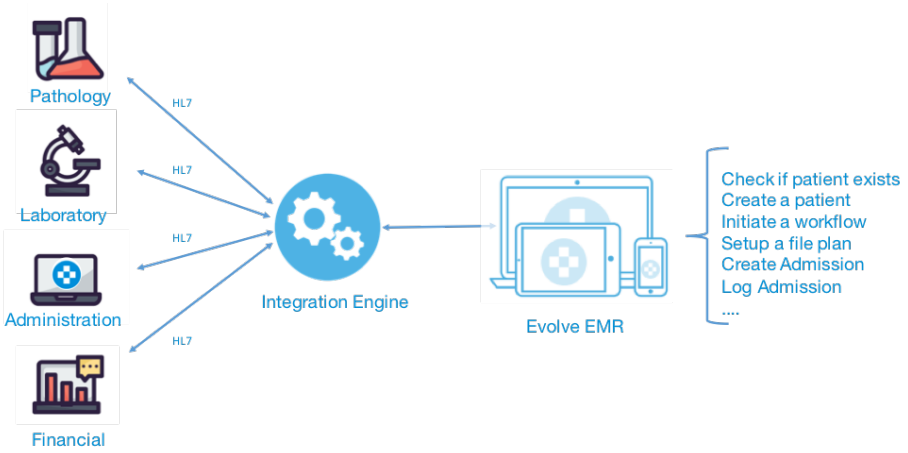

Figure 1: HL7 feeds and typical actions initiated within Evolve EMR upon message receipt

The following messages are used in this study: patient admission (A01), patient transfer (A02), patient discharge (A03), pre-admit (A05), cancel discharge/end visit (A13), add person information (A28), update person informa-

tion (A31), cancel pre-admission (A38) and merge patient (A40).

Evolve Integrated Care [25] (Evolve IC) has a different focus to Evolve EMR, in that its aim is to automate patient care pathways across many teams and organisations. Evolve IC is a cloud based, multi-tenant platform that collates information from a number of systems, such as EMRs (e.g. Evolve EMR), Ambulatory systems, Primary Care, Laboratory and Clinical systems. Evolve IC provides a central repository, which brings together information from each provider organisation, using Fast Healthcare Interoperability Resources (FHIR) [23], as illustrated in Figure 2. FHIR is the successor to the HL7 v2 messaging standard, and is primarily based upon the Json open standard [27]. Data in Evolve IC is also stored in the FHIR format.

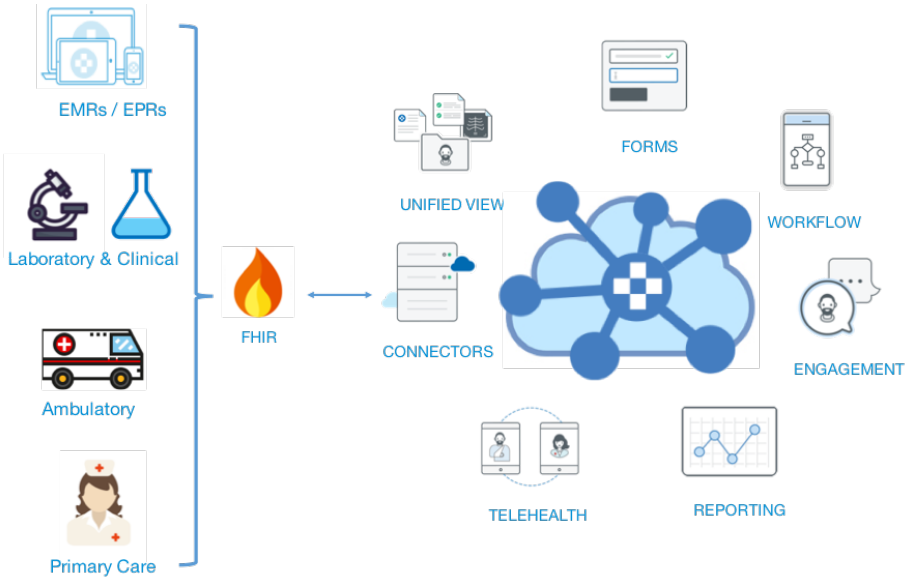

Figure 2: Evolve IC Platform, with associated functionality

3.3. Machine Learning and Anomaly Detection approaches

As highlighted in Section 1, ML is a technique that appears well aligned to cyber incident detection in an EMR system. ML can operate on large quantities

of data, and can detect anomalies and patterns within the data which would be impractical using manual review or rule based approaches.

Machine Learning has 3 main branches [28]:

- **Classification:** when given a new data point, determine how it relates to a series of existing defined data points or 'categories'. Classification may be binary; either the observed value belongs to one of two categories, or it may be multiclass, whereupon it can be assigned into multiple categories.
- **Clustering:** clustering operates by identifying groupings of related data points. For example, given a data set of IP addresses which accessed `www.bbc.co.uk`, they could be logically grouped by originating country.
- **Regression:** when given a series of previous data points, apply an algorithm to determine or predict a future value. For example, if we know a set of house prices for an area, linked to the size of house (number of bedrooms, bathrooms etc.), then the price of a new house with similar, or additional features, should be predictable.

Against the above types, machine learning can operate in a Supervised or Unsupervised manner.

Supervised Learning occurs when access to labelled training data is available. A label is simply a known identifier against a set of data. For example, in a series of examination results, labels could be assigned to the exam subject, score and the gender of the examination candidate. Classification and Regression tasks are examples of Supervised Learning.

Unsupervised Learning occurs when no labels are available for the dataset. Imagine a series of facial biometric points taken from a large volume of CCTV images. These points have no labels, yet given suitable training, an unsupervised algorithm may be able to determine a young person from an old person, or even a subject's ethnicity. Clustering is a typical form of Unsupervised Learning.

Analysing the ML classifications identified by Ahmed et al. in [29] (Figure 3), we identify that the Classification and Clustering categories of machine learning

algorithms appear best suited for cyber incident detection in an EMR system.

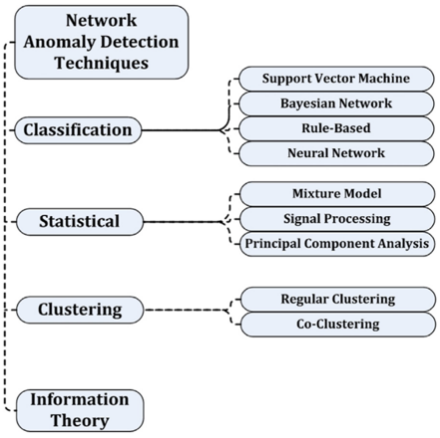

Figure 3: Anomaly detection classification [29]

Statistical and Information Theory categories of algorithms, as shown in Figure 3, are excluded from further consideration as possible detection mechanisms, as the study is not focusing on signal encoding or data compression techniques, for which these algorithms are best suited.

3.4. EMR System cyber incident detection - supervised or unsupervised learning?

Given sufficient data, Unsupervised Learning appears extremely powerful. The often manual and laborious task of labelling data is not required, as the algorithms can typically discern unknown or unexpected patterns or anomalies in the datasets, without observer intervention.

In a clinical setting, however, having access to large quantities of data can be problematic due to ethical concerns around patient information. Indeed, this issue has already arisen; the UK Information Commissioner Office (ICO) [30] stated that the Royal Free London NHS Foundation Trust did not comply with the Data Protection Act when it provided sensitive medical data of around 1.6 million patients to Google, as part of a clinical safety initiative.

Despite the data access issue, there are some examples of Unsupervised Learning applied to Cyber Security incident detection. For each of the examples, we briefly describe the solution and discuss the challenge for adoption of the solution for the EMR system. This leads to our specific algorithm selection for cyber incident detection in EMR systems introduced in Section 5.

In [31], Pan et al. evaluate the feasibility of unsupervised learning for web attack detection based on the Robust Software Modelling Tool (RSMT). RSMT monitors and characterises the runtime behaviour of web applications by injecting shims into the binaries of web applications and then profiling the control and data flow from the program components as they execute under normal conditions.

The approach uses two web applications as a testbed; a video management application and a compression application, each of which is subjected to a number of OWASP Top 10 attacks, namely SQL Injection, Cross Site Scripting and Object Serialization attacks. A stacked de-noising auto-encoder algorithm [32] is used to generate a low-dimensional representation of the raw features with unlabelled request data. Anomalies (or attacks) can then be recognised by computing the reconstruction error of the original request data.

Applying this method to the problem space of Cyber Incident Detection in EMR systems is potentially feasible, but it would require that the complete operation of all system calls be profiled to determine what normal use looks like. This profiling could only really be applied to other Trust systems connecting into the EMR, as they would typically follow repeated forms of operation in their daily use.

More specifically, applying this method to the detection of malicious access of patient records by Trusted System Users would require significant redesign, as the method focuses on the detection of technical attacks against the system (SQL injection or XSS attacks) rather than the detection of valid system credentials being used to access patient records under specific abnormal scenarios. However, two findings from [31] are relevant; firstly, the confirmation that collecting labelled training data in large scale production systems can be difficult,

and secondly, identification that the Autoencoder algorithm requires about 5000 items of unlabelled training data to achieve performance. This particular finding highlights the difficulty in obtaining sufficient data in a Healthcare setting.

In [33], Moradpoor et al. use Principal Component Analysis (PCA) in conjunction with a Self-Organising Map (SOM) for insider threat detection. A commercial User Behaviour Analytics product named ZoneFox was used as the basis for testing. Six scenarios were then constructed to simulate insider threats ranging from a privileged user data breach for a temporary staff member, to system data theft and access of sensitive folders.

The study principally focuses on improvements in existing detection ability by applying PCA to input datasets before further processing by SOM. From the perspective of cyber incident detection in an EMR system, the approach appears targeted at file based systems, which is somewhat applicable to Evolve EMR as medical records are typically rendered as PDFs and stored within a centralised and restricted file-share. These medical records are visualised within the EMR client, along with patient metadata. Therefore, file access audit records could be used to facilitate detection to some extent.

The gap in the above approach is that it does not appear to cover application specific usage, such as a user accessing a record within the EMR system that does not necessarily have an associated PDF file.

In [5], local outliers are used to detect incidents of insurance fraud via excessive medical treatment in a Chinese city. Three datasets are used which represent medical insurance claims (over 3228 hospitals, covering 221,000 patients), hospitalisation details of patient stays (50 million) and information on insured residents of the unnamed city (around 690,000 records).

Features were extracted from these datasets that indicate excessive medical treatment, excessive medical visits and disproportionate claims in relation to hospital stays. The study confirms that the application of ML is effective in finding cases of fraud, but what is most enlightening is the access to such substantial datasets. It is unlikely that access to such wide ranging data sets would be granted in UK or US hospitals, due to patient privacy concerns and

legislation such as GDPR [34]. Nevertheless, the study provides insight into the potential incident detection capability.

Based on the review of the suitability of approaches, and given the limitations of access to datasets sufficient to enable unsupervised learning, this study progresses with supervised learning techniques.

4. Problem Definition

For the purposes of this study, a cyber incident is defined as an event that impacts the confidentiality, integrity or availability of an EMR system. This definition aligns with [35, 36] and is used to shape the scope of the research and to formulate specific, targeted incident detection scenarios against each of these attributes. 'Incident Detection' is one aspect of the problem space. However, the solution(s) may be able to successfully detect an incident, but be infeasible, impractical or simply not perform well enough to be used in a clinical production setting. Therefore, the problem definition is fully stated as follows: *How feasible, practical and performant is cyber incident detection against an EMR?* and further detailed in Table 1.

Table 1: Problem Definition

Feasibility	Data	Does HL7, FHIR or Evolve IC audit data support the use case scenario? What auditing practices need to be used to support the detection?
	Tools/Algorithms	Do suitable tools and algorithms exist to support the detection approach? How quickly can anomalies be detected in real time? Is a ML 'pipeline' required to prepare data?
Practicality	Difficulty	How difficult is it to understand and apply the machine learning algorithms? Are specialist skills required?
	Repeatibility	How easy is it to repeat the training/testing process against a real EMR product? Can new anomaly types be added to the detection mechanism?
Performance	ML performance will be assessed against:	$Accuracy = \frac{TP+FP}{TP+FP+TN+FN}$ $Precision = \frac{TP}{TP+FP}$ $Recall = \frac{TP}{TP+FN}$ (*TP: True Positive, TN: True Negative FP: False Positive, FN: False Negative*)

4.1. Threat Model

Figure 4 presents the architecture of a typical EMR system.

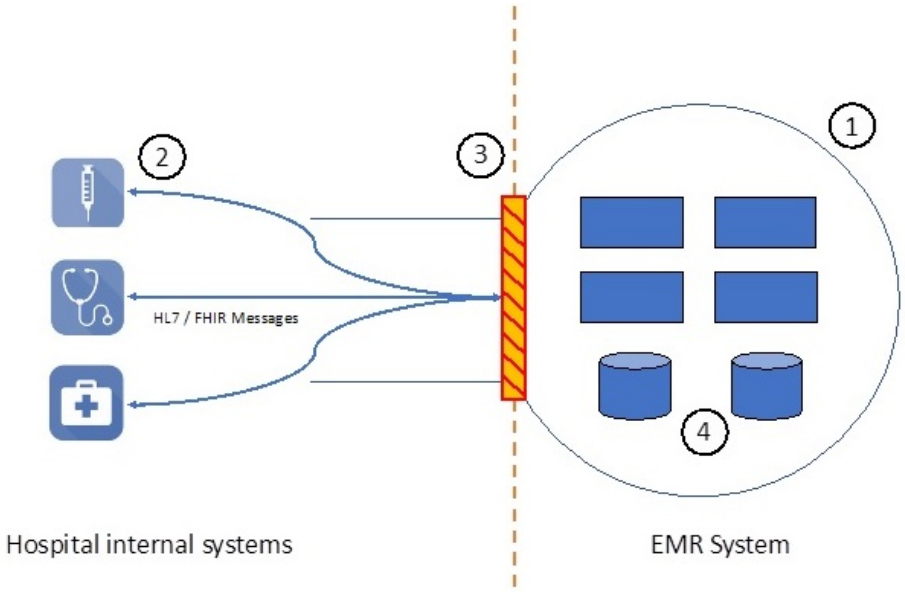

Figure 4: EMR System Threat Model

1. The EMR may be deployed on-premise, where it runs on infrastructure owned and operated by the Hospital Trust, or it may be Cloud-based, whereby a Cloud provider such as Amazon Web Services or Microsoft Azure is used to provide the underlying virtual machines and networking infrastructure. In both cases, the 'bubble' (in Figure 4) for the EMR is reasonably well protected. In the case of the EMR using the hospital infrastructure, a separate network subnet is used to host the EMR. In the Cloud scenario, a virtual private network protects the EMR from access by normal internet traffic. This isolated network approach protects the EMR from a number of adversaries, such as automated internet attacks, script-kiddies or organised criminal gangs looking to exploit open systems.
2. A number of connected systems send traffic into the EMR. This traffic ranges in variety, from HTTP/S to access the EMR Web-Client, to

HL7/FHIR messages that instigate patient admission, discharge or transfers within the EMR. Typically, these connected systems are 'Trusted', meaning that they may route traffic to the EMR over TLS, and/or be whitelisted and approved to send such traffic. These types of systems range from Pathology or laboratory systems, to EMR Client software. The use of these connected systems is restricted and their access is typically secured using domain credentials. A clinician or lab assistant would need to authenticate to the system, whereupon role-based-access (RBAC) control would restrict what operations they can carry out, and what information they can view, as defined in Section 2.

3. A firewall appliance typically blocks all non-essential traffic from entering the EMR subnet. This defensive measure is needed to mitigate threats such as the WannaCry worm, which spread rampantly throughout NHS Trusts in May 2017 [37].
4. Within the EMR, there are typically a number of components such as webservers, file-shares and databases. As with item 2, RBAC is extensively employed throughout the estate, to restrict access to these elements to only trusted system users.

A threat model to the EMR system can be extrapolated based on this assessmnet; External means outside the Trust infrastructure, and Internal means within the Trust systems / domain. The EMR system threat model is summarized in Table 2.

Table 2: EMR System Threat Model

Actor	**Source**	**Threat**	**Likelihood**	**Mitigation**
Organised Criminal Gang/ Hacktivist/ Script-Kiddie	External	Denial of Service/ Theft of Patient Data	Low	If the EMR is hosted on the Trust's subnet, no internet access is possible. In a properly secured cloud deployment, likelihood of access is low.
Bot/Worm	Internal/ External	Ransomware	Medium	An epidemic such as Wannacry spread from machine to machine. In the case of on-premise or cloud deployment, a properly configured firewall would mitigate this threat.
Trusted User (Clinician)	Internal	Theft of Patient Data	Medium	Role-based access controls and auditing are typically used to mitigate against this threat. However, the volume of data is so great that detection is difficult.
Trusted System i.e. PAS	Internal	Denial of Service	Medium	IP whitelisting secures the traffic route, but does not mitigate against the volume of traffic coming from a Trusted system.

In this work, we focus on the last two medium likelihood internal threat vectors described in Table 2. The increasing cyber threat from insiders classified as masqueraders, traitors, and unintentional perpetrators is clearly outlined in [16].

4.2. Confidentiality Scenario

A breach in confidentiality is the unauthorised access or disclosure of information. In an EMR setting, this means that patient data is being accessed by a person who is not authorised to do so. It is assumed that RBAC is implemented as a standard EMR security measure and, as a result, the EMR system cannot be readily accessed by anyone without the requisite permissions to do so.

Therefore, the scenario is one in which a known and trusted user of the system is acting in a malicious or inappropriate manner, or an adversary has obtained the credentials of someone with valid access to the EMR system, and is using those credentials to access patient data. Either situation leads to a breach in confidentiality as the patient data is not protected from viewing. However, there is one exception to these conditions; when an emergency has occurred and a clinician needs access to records of a patient in order to aid them. In this case, the access attempt is valid. This particular situation is known as an emergency 'break-glass' event.

To determine patient access patterns, statistics were obtained from two UK hospitals over the period of a single day (shown in Table 3). The statistics show the number of active users within the hospital, and the number of patient record views. From these figures, the average number of access attempts during a 24-hour period, per hospital, is deduced.

Table 3: Hospital medical record viewing figures (per 24hrs)

	Users	Patient Views	Average Access
Hospital 1	1073	61971	57
Hospital 2	672	19891	29

This average access value is used to shape the synthetic data requirements.

Unfortunately, it is not possible to derive a more detailed segregation of user types or groups, for example, a breakdown of access attempts by nurse, clinician or administrative user groups, who may have different viewing averages per group. This is an informal restriction on sharing patient related information as a result of the UK Information Commissioner Office (ICO) statement [30] in July 2017, as described in Section 3.

The confidentiality scenario is defined as described in Table 4:

Table 4: Confidentiality Scenario

Scenario	Identify anomalous read events by clinician users against patients within a 24hr period. Anomalous read events occur when a user accesses a patient record when they have no relationship to the patient and when no break-glass event exists.
Relationship	A relationship is defined by the presence or absence of appointments, observations or encounters records linking the patient and the clinician user.
Break-glass Event	A break-glass event is when a clinician legitimately accesses a patient record in an emergency situation. During this event, the reason for accessing the patient's record is recorded against the access attempt.
Inputs	Patient Access Statistics, Evolve IC audit events, FHIRbase data
Platform	Evolve IC
Expected Detection Output	The detection system should be able to identify anomalies in patient record access by a clinician.

4.3. Integrity Scenario

A breach in integrity is the unauthorised modification or destruction of data. In an EMR setting, this means changing or destroying patient information or any related records. For example, a clinician may prescribe '100 mg Neurofen'

to a patient, but the medical dispensation record is maliciously altered and the patient is subsequently given '1000 mg Neurofen', or the patient record may be deleted in its entirety. Upon investigation into this area, a number of substantial challenges arose when constructing a valid ML-based Integrity scenario. These are recorded here for completeness. However, as a result of the limitations, the integrity scenario is not included in the research results.

1. A single change to a single field in a patient record could impact its integrity. A malicious user, or someone who has stolen a trusted user's credentials, may choose to only carry out a single change against a single patient record and do no more. This situation is very difficult for machine learning algorithms to detect, as they are based on evaluating patterns in amassed data. This finding was highlighted in [21].
2. No medical consultation was available to the authors to understand whether a breach of integrity has occurred against medical data. For example, a medical record could detail the prescription of 100 mg Paracetamol, but be altered by a registered user, to be 100 mg Ibuprofen. The drug types would differ, but the medical treatment for the patient would be somewhat equivalent. This type of incident would be very difficult to detect without expert knowledge.
3. No statistical information was available from a real hospital on the number of updates or deletions to patient data by user groups, leading to a lack of comparison data. Indeed, this statistical information would have been too course-grained to enable any valid investigation.

4.4. Availability Scenario

An availability incident is the disruption of access to, or use of, information or an information system. Typically, this disruption is caused by a DoS attack, against external internet facing systems. Evolve EMR is an internal system, but it is connected to other messaging systems such as patient administration or laboratory systems and these could act as potential DoS vectors.

For a major UK Hospital (referred to as Hospital 3), the HL7 message volumes for a 6-month period covering September 2016 to February 2017 were obtained. From this data (shown in Figure 5), it is identified that the normal volume of A31 'Update person information' messages, falls within the 43k to 49k messages per month range, with A05 'Pre-admit patient' messages falling within the 64k to 77k range. However, in September 2016, A31 messages surged to 240,583 messages, and in October 2016 a similar surge, to 133,109, occurred for A5 messages. These surges caused non-trivial impact throughout downstream systems, such as Evolve EMR. The additional A31 and A05 messages were found to be caused by duplicate messages being issued from the upstream patient administration system. Note that for A31 messages, the surge continued into October whereupon the backlog of messages was eventually remedied.

		Sep 16	Sep 16	Oct 16	Oct 16	Nov 16	Nov 16	Dec 16	Dec 16	Jan 17	Jan 17	Feb 17	Feb 17		Average
ADT_A01	Admit/visit notification	13,114	3.6%	11,529	3.6%	11,955	6.9%	11,589	7.4%	11,782	6.6%	10,488	6.5%	11,743	5.2%
ADT_A02	Transfer a patient	4,674	1.3%	4,889	1.5%	4,749	2.7%	5,063	3.2%	5,127	2.9%	4,785	3.0%	4,881	2.2%
ADT_A03	Discharge/end visit	11,300	3.1%	11,830	3.7%	12,132	7.0%	11,925	7.6%	12,019	6.7%	10,777	6.7%	11,664	5.2%
ADT_A05	Pre-admit a patient	71,903	19.7%	133,109	41.4%	73,377	42.2%	64,224	40.8%	76,910	43.0%	68,855	42.6%	81,396	36.0%
ADT_A13	Cancel discharge/end visit	172	0.0%	197	0.1%	177	0.1%	178	0.1%	198	0.1%	178	0.1%	183	0.1%
ADT_A28	Add person information	5,336	1.5%	5,311	1.7%	6,172	3.5%	5,794	3.7%	6,146	3.4%	5,752	3.6%	5,752	2.5%
ADT_A31	Update person information	240,583	66.0%	125,605	39.1%	48,968	28.2%	43,641	27.7%	47,361	26.5%	45,152	27.9%	91,885	40.6%
ADT_A38	Cancel pre-admit	17,240	4.7%	28,847	9.0%	16,363	9.4%	14,806	9.4%	19,122	10.7%	15,620	9.7%	18,666	8.3%
ADT_A40	Merge patient - patient identifier list	144	0.0%	107	0.0%	24	0.0%	70	0.0%	69	0.0%	37	0.0%	75	0.0%
		364466	100.0%	321424	100.0%	173917	100.0%	157290	100.0%	178734	100.0%	161644	100.0%	226,246	100.0%

Figure 5: HL7 Monthly message volumes showing A31 and A05 surge

The effect the surge of messages had on Evolve EMR was equivalent to a DoS attack. Given these inputs, the Availability scenario is defined as described in Table 5:

Table 5: Availability Scenario

Scenario	Identify anomalous HL7 A31 messages that are coming from an upstream PAS system. Messages are sent throughout the day, from many systems. Therefore, the DoS detection approach needs to be as immediate in nature as possible in order to mitigate the effects of the incident.
Inputs	HL7 message statistics from Hospital 3 are used to create test datasets. 10% of monthly volumes are used, to reduce the amount of testing data required. The ratio of messages is maintained to ensure that they represent the distribution of messages observed in Hospital 3 when the outage occurred, albeit on a smaller scale. A dataset is produced that represents a normal month, based upon Feb. 2017 volumes, and one that represents an anomalous month, based upon Sept. 2016. A further combined dataset is created that combines both these months.
Platform	Evolve EMR
Expected Detection Output	It is expected that a level shift change in the volume of A31 messages will be detected.

5. Methodology

For the confidentiality and availability scenarios defined in Section 4, the suitability of specific ML and TS algorithms have been explored.

5.1. Confidentiality Scenario - ML Classification

Figure 6 presents a view of possible ML algorithms available for incident detection. As identified in Section 3, classification and clustering algorithms are suitable for cyber incident detection in an EMR. Classification is a type of supervised ML while clustering is a type of unsupervised ML. The 'blue boxed' algorithms have been selected for the Confidentiality Prototype. The reasoning for this decision follows in the text below.

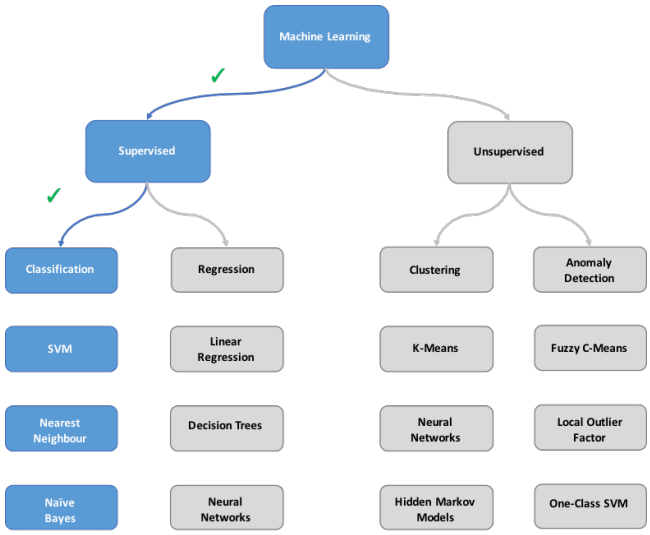

Figure 6: Machine Learning Algorithms

Supervised techniques depend on the availability of a training dataset, with labels to differentiate normal observations from abnormal or anomalous observations. A label is simply an indicator that signifies whether a particular row in a training dataset is normal or anomalous. The process of supervised learning is one of gradually training and correcting the detection model over time, to

obtain an acceptable level of performance. In contrast, unsupervised learning leaves the algorithm to derive meaning from the testing data itself with no training phase interaction or correction applied to the model. Unsupervised learning typically derives meaning from the data that is not immediately obvious to human observers. Further detail on unsupervised learning has been provided in Section 3.

In the case of the Confidentiality scenario, all data has been synthesised as part of this study due to the restrictions on obtaining real audit information from a hospital, and it can be labelled accordingly. Therefore, supervised classification is determined to be more appropriate for the Confidentiality Scenario. The classifications are known upfront, namely: 'Normal' (clinicians with expected observations, appointments or encounters with a patient, who subsequently access the patients record in a normal or emergency break-glass scenario), and 'Anomalous' (those who do not meet these conditions). Support Vector Machines, Nearest Neighbour and Naive Bayes are all types of classification algorithms. These three candidate algorithms will be assessed as part of the Confidentiality Prototype to determine the best fit, as recommended in [38].

Note that although it is a supervised technique, regression is not applicable to the goal of detecting a confidentiality incident, given the scenario definition i.e. we are not aiming to use the data to predict the potential future number of malicious insider users.

5.2. Availability Scenario - TS Anomaly Detection

Time Series (TS) data is a set of observations, in sequence, and usually at fixed intervals. TS anomaly detection is relevant to this study as it can be applied to the HL7 message flow from a PAS to an EMR system, specifically to look for evidence of a significant increase in message flow which could be the onset of an availability incident. There are three types of TS anomaly profile:

- *Point:* A specific event that is distinguishable from the rest of the dataset. For example, a single excessive payment on a credit-card statement.

- *Contextual:* The anomaly occurs at an unexpected interval of time. For example, a high temperature observation occurring in the middle of winter.

- *Collective:* The anomaly occurs in respect to the rest of the entire dataset being measured. For example, when reversing a vehicle, some vehicles may emanate a beep to represent the distance from the rear of the vehicle to any objects located behind it. If no movement is performed, the beep frequency remains constant, however if the vehicle rapidly moves closer to the object, the beeps increase to a new frequency. This change is also known as a mean shift or level shift [39].

Given these categories of anomaly, the *Collective* profile aligns best with the profile of a DoS availability incident; a surge in HL7 messages from a PAS to an EMR system would effectively be a level shift event.

Note that in IT systems, bulk data loading or performance testing may have the same impact as a level shift. However, these events are typically performed under known and controlled scenarios. An availability incident is an unexpected level shift.

A number of TS algorithms have been considered for applicability in this study: Season-Trend decomposition (STL), Classification and Regression Trees (CART), Autoregressive Integrated Moving Average (ARIMA), Exponential Moving Average (EMA), and Long Short-Term Memory (LSTM). EMA was selected for further testing for the following reasons:

- Strong evidence of practical implementation e.g. LinkedIn has been running an EMA based library named "Luminol" for over a year against all LinkedIn pages and apps, for the purpose of detecting surges in web traffic;

- Ability to integrate the 'Luminol' library [40] to the Availability prototype framework, and

- Limited data manipulation requirement.

6. Test Environment

The following section details the prototype construction for both the confidentiality and availability prototypes. In addition to the selection of appropriate detection algorithms for application to the defined problem, significant effort has been invested in generation of the data sets against which to test the algorithms. Synthea [41], a Synthetic Patient Population Simulator, that is used to generate the synthetic patients for the SyntheticMass [42] tool, was used to generate approximately 36,500 patients. SyntheticMass contains a million realistic, but fictional, residents of the state of Massachusetts. The synthetic population statistically mirrors the real population in terms of demographics, disease burden, vaccinations, medical visits, and social determinants. The specific data generation for each prototype is detailed in the next sections. The complete data sets and implementation code used to produce the results in this paper are available at [10].

6.1. Confidentiality Prototype

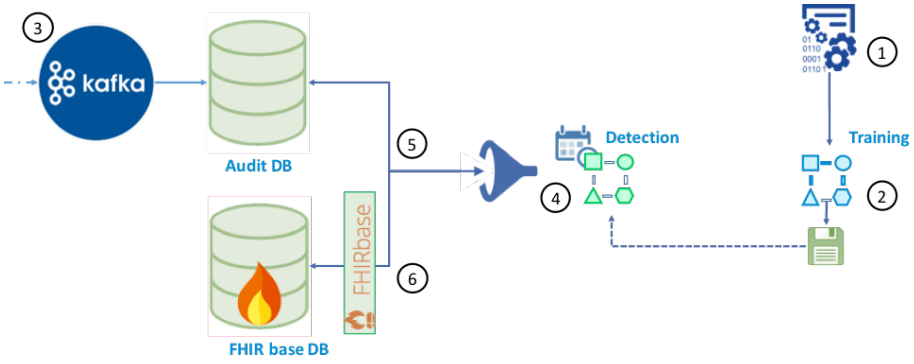

Figure 7: Confidentiality Prototype Design

Figure 7 illustrates the design used for the Confidentiality Prototype. There are two phases in the operation of the prototype; the first being the training and persistence of the ML models (1,2 in Figure 7), and the second being the

detection application using the trained models to detect incidents (3-6 in Figure 7).

Step 1: Data Generation Pre-prepared data is required in order to train the ML models. This data is generated based on the structure of the Evolve IC audit_event table, and the FHIR database which stores appointment, observation and encounter information. Synthea was used to generate patient information such as names and addresses, and this was used to simulate the representation of patient records within the FHIR Database. Other records, such as appointments, observations and encounters were linked to these patient records, and key attributes shared between records (i.e. join fields).

Step 2: Model Training Within the Training Application, each of the three selected ML models (Support Vector Machine, Nearest Neighbour and Multinomial Naive Bayes) are trained, validated and then persisted to disk using the Python Pickle Library [43].

Step 3: Kafka Event Stream Apache Kafka [31] is a distributed streaming platform. Kafka collates all events that occur across Evolve IC services. These events are ultimately persisted to the audit_event table in the Audit database. Simulating a significant volume of clinician user activities such as opening a patient's record using the Evolve IC interface is impractical, therefore audit events were simulated by creating the equivalent audit_event rows.

Step 4: Confidentiality Detection Application The detection application is run on a scheduled basis (once every 24 hours). The application begins by reloading the previously trained models from storage (as described in Step 2).

Step 5: Audit DB Query The next action within the detection application is to query the audit database and retrieve relevant rows from the audit event table. The key part of the query is to obtain 24 hours worth of events.

Step 6: FHIR Queries Each row retrieved from the audit_event table signifies the access of a patient record by a clinician. The application then aggregates the related number of appointments, observations and encounters from the FHIRbase database. The individual counts of appointments, observations

and encounters, plus the special_action flag, are all passed to each ML model to obtain a class prediction. Note that the special_action flag signifies a break-glass emergency event.

The application's output is then collated and used to determine the accuracy of each of the models. The confidentiality test results and analysis are presented in Section 7.

6.2. Availability Prototype

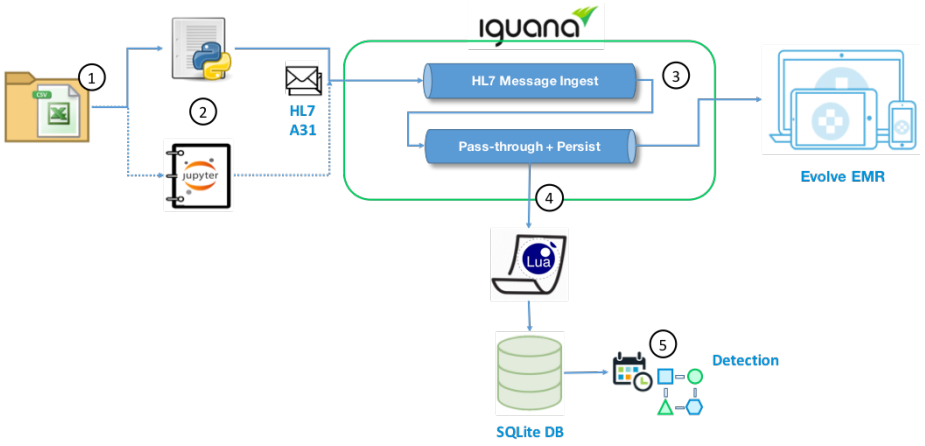

Figure 8: Availability Prototype Design

Figure 8 shows the design used for the Availability Prototype. Again, there are two distinct phases in the operation of the Availability Prototype; the first being HL7 Message Generation (1,2 in Figure 8), and the second being message processing and detection of a level shift in message throughput (3-5 in Figure 8).

Step 1: Patient Data Generation The base content for the HL7 messages is patient data. For the Availability Prototype, Synthea was used to generate a CSV file of patients for use as a base for HL7 message generation. A python script was created to construct HL7 messages using the base data from this CSV file. This script simply constructs the HL7 message format, and at the various

message segments which require patient data, the relevant data fields from the Synthea CSV file is inserted. As previously noted, the complete data sets and implementation code are available at [10].

Step 2: HL7 Message Generation Each row in the CSV file is used to craft an HL7 message from the data.

Step 3: Siphoning Messages within Iguana Iguana is the integration engine that processes the messages from a PAS into the EMR system. Two channels were configured within Iguana, namely:

1. Read HL7 files to Queue: This step loads HL7 messages from a specified folder and persists the messages onto an Iguana queue. This action simulates a PAS sending messages to Evolve EMR.
2. Queue to DB: Takes each individual message and persists the message to a SQLite database.

Note, Figure 8 shows a further connection onward to Evolve EMR. This would be the normal route of HL7 messages. The availability incident detector would simply siphon a copy of the messages for processing (via an additional channel configuration in Iguana).

Step 4: Persisting to Storage The contents of each HL7 message is persisted to storage (a SQLite database). This is required to provide the Luminol library with previous telemetry data in order to ascertain if a level shift has occurred. Persistence is achieved by including a lua script within the integration engine.

Step 5: Availability Incident Detection The Availability Prototype is a standalone application which operates against the SQLite database. It begins by following a job schedule that is defined using the Python schedule library [44]. SQLite is queried to retrieve all persisted HL7 messages fields. The query groups messages into fixed intervals (e.g. 10 seconds) by using a database field that represents the time the message hit Iguana. Fixed intervals are required by the Luminol EMA library to enable statistical analysis to be carried out. The interval batch is then passed into the Luminol Anomaly Detector. The detector

type is explicitly set as exp_avg_detector the exponential average detector within the Luminol libraries configuration.

The final aspect of the Availability Incident application is to call the get_all_scores and get_anomalies methods of Luminol, which retrieves the anomaly scores against each observation, and any that are greater than the threshold, respectively. The score is an indicator of whether an anomaly has occurred within the lagging window for the exponential moving average algorithm. If an anomaly is detected, the application will quit. This is in line with the detection of a level shift, i.e. there is no further need to continue. The complete scores across the full stream of HL7 messages are then used to visualise what anomalies have occurred. The availability test results and analysis are presented in Section 8.

7. Results and Analysis - Confidentiality Scenario

7.1. Confidentiality Scenario Results

Two steps of testing were applied for the Confidentiality Scenario:

1. Initial Model fitting and testing using a training file. This step fits the models using labelled data and then tests the results (with a portion of the training file).
2. Testing against the Evolve IC instance. Retrieve all required record counts (Appointment etc.) from Evolve IC and use this data to obtain a prediction.

7.1.1. Step 1 - Model Fitting Results

K-fold Cross Validation [45] was used to train the models in order to maximise exposure to training data. First, the training file was split into Training, Validation and Test sets. 34% of the training file was retained for use as a Test Set with the remaining 66% being used for Training and Validation. The Test Set size meets the recommendation outlined in [46]. The aim of using K-Fold is to rotate and train/validate a model against each fold in turn, before being exposed to the unseen Test set. In this case, 684 rows of the 2010 rows form the

Test Set, with the remaining 1326 rows being used for training and validation. This is illustrated in Figure 9.

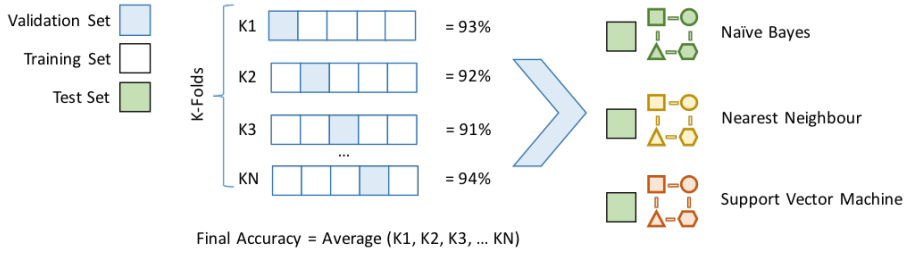

Figure 9: K-Fold Cross Validation [45]

The results of the K-Fold tuning are presented in Figure 10. By validating the model after each fold in turn, we can measure its average accuracy after N-folds. Figure 10 shows that 7-folds provides the best detection accuracy, and the highest average across all models. 7-folds is therefore used as the default parameter for Cross Validation.

	Accuracy			
K-Folds	KNN	SVM	MNB	Average
3	0.9729	0.9864	0.7044	0.8879
5	0.9713	0.9925	0.7059	0.8899
7	0.9864	0.9880	0.7466	0.9070
10	0.9834	0.9910	0.7105	0.8949

Figure 10: K-Fold Tuning against 3,5,7 and 10 folds

The next stage in fitting is to run each model against the Test Set. This Test Set is effectively new and unseen data for all the models. The precision and recall results obtained for KNN, SVM and MNB when testing against the Test Set are provided in Table 6. Precision is the ability of the classifier not to label as positive, a sample that is negative. Recall is the ability of the classifier to find all the positive samples.

As shown in Table 6, when tasked with predicting values in the Test Set, KNN and SVM achieve significantly better results than MNB in Precision and

Table 6: Precision and Recall from Validation Testing

	KNN		SVM		MNB	
	Precision	Recall	Precision	Recall	Precision	Recall
Anomaly	0.87	0.99	1.00	0.94	0.60	0.42
Normal	1.00	0.93	0.97	1.00	0.75	0.86
Avg./Total	0.95	0.95	0.98	0.98	0.70	0.72

Recall, i.e. their ability to correctly find and label anomalies.

7.1.2. Step 2 - Evolve IC Testing

Step 1 covered training, validation and testing of the 3 models. Step 2 tests the models against a working development instance of Evolve IC, using data that has not been seen by any of the algorithms. Each test was carried out three times. The results are presented in Table 7.

Table 7: Confidentiality Scenario Test Results

	KNN		SVM		MNB	
	Anomalies	Normal	Anomalies	Normal	Anomalies	Normal
Test C1	4	2006	0	2010	22	1988
Test C2	5	2005	1	2009	22	1988
Test C3	3	2007	7	2003	22	1988

Test C1: For the first test, zero anomalies were present in the Evolve IC instance. This test is to ensure that the application is not detecting any false positives in the dataset. The output was identical for each of the 3 test runs. SVM performed as expected and did not detect any anomalies. Analysis of the KNN output indicated that the 4 anomalies detected were against a standard pattern of 1 appointment, 0 observations, 1 encounter and no break-glass event. When examining the training set, only a limited number of examples of this permutation were present (3 items from a training set of 2010 rows). From this finding it can be inferred that KNN was not exposed sufficiently

to this permutation during cross validation and that the training set needs to be enlarged to provide sufficient coverage of all permutations. The MNB classifier detected 22 false positives. When examining the training set, it was discovered that for the records flagged as anomalies by MNB, all had greater than 5 observation records. Note that 5 was the upper value that was used in the Training file for all appointments, encounters and observations. From this finding it can be inferred that MNB struggles to classify data that it has not seen during training.

Test C2: A single anomaly was added to clinician User000007. This was achieved by ensuring that no appointments, observations, encounters or break-glass actions are recorded in the audit_event table. 3 test runs were carried out. The output was identical for each of the 3 runs. Again, SVM performed as expected and detected a single anomaly. KNN detected the anomaly, and included the 4 false positives from the previous test. The MNB classifier continued to exhibit incorrect readings and failed to detect the single true anomaly.

Test C3: In this test, 3 different anomaly types were added to Evolve IC against different clinicians. Anomaly types 1, 5 and 10 from the classification matrix in Figure 11 were used. Note that 'Y' denotes that 1 to 5+ records exist, whereas 'N' denotes that no data is present. [1]

[1] It should be noted that this classification has been developed and proposed in collaboration with an EMR specialist but has not been formally approved by a hospital administration.

ID	Appointments	Observations	Encounters	Break-Glass	Class	Description
1	N	N	N	N	Anomaly	With no Appointments, Observations, Encounters or Break-Glass events, there is no relationship between the Clinician and the patient, and no emergency has occurred.
2	N	N	N	Y	Normal	A Break-Glass event has occurred.
3	N	N	Y	N	Anomaly	The Clinician has met the patient but no appointments exist for a future consultation.
4	N	N	Y	Y	Normal	The Clinician accessed the patient records in an emergengy situation
5	N	Y	N	N	Anomaly	The Clinician has observed the patient but no appointment exists for a future consultation.
6	N	Y	N	Y	Normal	Break-Glass event
7	N	Y	Y	N	Anomaly	The Clinician has met and observed the patient but no appointment exists for a future consultation.
8	N	Y	Y	Y	Normal	Break the Glass event
9	Y	N	N	N	Normal	Future appointment
10	Y	N	N	Y	Anomaly	The Clinician has an appointment with the patient, but has accessed their record via a Break The Glass event.
11	Y	N	Y	N	Normal	The clinician has met the patient and appointments exist
12	Y	N	Y	Y	Normal	The clinician has met the patient and has had to access their file in an emergency situation
13	Y	Y	N	N	Normal	The clinician has observed the patient and has future appointments to do so.
14	Y	Y	N	Y	Normal	Emergency access
15	Y	Y	Y	N	Normal	Normal hospital visit
16	Y	Y	Y	Y	Normal	The clinician has met the patient and has had to access their file in an emergency situation

Figure 11: Confidentiality Prototype: Classification Matrix

The output was identical for each of the 3 test runs. As previously, SVM performed as expected and detected all anomalies. KNN detected 3 anomalies with the 4 false positives from Test C1. The MNB classifier continues to exhibit incorrect readings.

7.2. Confidentiality Scenario Peformance

Training the models took only a number of minutes against a dataset of 2010 records. Across all three tests against the Evolve IC instance, the speed of the detection application was recorded to be between 64 and 78 seconds, which is well within the parameters required for 24 hr operation.

The performance of this approach will be determined based on the following factors:

- Number of audit records to process, which is typically a measure of the number of clinicians and nurses using the EMR system.
- The number of features being monitored. In this study, only 4 were used.

The confidentiality prototype ran sequentially against the instance of Evolve IC. It initially queried the audit database for records that indicated a patient record was accessed by a clinician, then retrieved aggregate counts of appointments, observations, encounters and break-glass events for each individual audit record linked to the same clinician.

This approach is certainly not the most optimised measure, and could indeed be improved by maintaining denormalised running count of patient, clinician, appointment, observation, encounter and break-glass events in a separate database table. This mechanism would require greater engineering effort, but would have the positive effect of eliminating sequential queries whilst providing the means to more easily add new features for inclusion in any future ML processing.

7.3. Confidentiality Scenario Analysis

A number of observations are evident when applying Machine Learning to detect a Confidentiality incident against Evolve IC. Firstly, the prototype has

successfully detected anomalies during the Evolve IC test runs using Classification based ML algorithms. This result proves that cyber incident detection is possible for the Confidentiality Scenario.

With regard to the algorithms tested, the Support Vector Machine has obtained the highest accuracy during model testing, recording 98.94% during Cross Validation, and 100% when tested against Evolve IC over 3 tests.

The Scikit-learn framework [47] was used to implement the SVM, MNB and KNN algorithms, as it provides extensive documentation and an intuitive development process.

For the implementation of SVM, default hyper-parameters were used during the training and testing of the ML algorithm. Data was composed of a single training file that had been partitioned into separate training, testing and validation sets. A support vector machine constructs a hyperplane/set of hyperplanes in a high dimensional space, which can be used for classification, regression or other tasks. The Sci-kit implementation uses a kernel function to implicitly map data into a feature space where they are linearly separable.

Performance of the algorithm was extremely good using the Test Rig set up for the prototype. Training the base algorithm took less than 1 minute, and when applied to a real instance of Evolve IC, the detection aspect of the prototype completed in around 2 minutes.

To clarify, the size of the test file used in the confidentiality prototype is representative of Hospital 2, which has 672 Clinical users. Other hospitals may have significantly more users, others less so. Our recommendation based on this study is to use a separate data store (other than the production data store) against which to run the ML algorithms. This will ensure that the production system remains unaffected.

Furthermore, given the nature of the task, i.e. identifying access to patient records without permission, a scheduled job which runs once per day should provide a suitable response time-frame.

Reviewing the three algorithms, KNN displayed reasonable detection abilities. However, a specific combination of appointments, observations, encounters

and break-glass flag were consistently flagged as anomalous by this model. Upon investigation, it was determined that the Training Set did not have sufficient coverage of the specific permutation which was flagged consistently as anomalous by KNN. This is indicative that the Training Set needs to have equal representation of all Normal/Anomalous permutations to ensure that the model is adequately trained. The MNB performance was mediocre across all test runs, when compared to SVM and KNN. When the source of the anomalies was examined, it was discovered that MNB failed when dealing with observations, which were unseen during training; specifically those with > 5 observation counts. Given that the cross-validation results for MNB were also significantly lower than SVM and KNN and that MNB failed to detect any true anomalies across the 3 test runs, it is concluded that the MNB algorithm is unsuitable for cyber incident confidentiality breach detection.

In Section 4, the feasibility and practicality of the detection solution were also raised. In terms of applicability to a Clinical Setting, a question remains as to whether four variables (appointment, observations, encounters and break-glass) can provide sufficient evidence that a malicious act has been performed. Real-life often introduces unforeseen circumstances that would not be accounted for in a ML based solution. For example, a single observation on its own initially looks like an anomaly, but an observation can also be carried out during a patient death event, which would of course not necessarily have an associated appointment or encounter scheduled. Therefore, the potential exists for a clinician to be called in to perform a time-of-death observation against a patient, which would actually be a 'Normal' type event. Alternatively, a clinician really could be accessing a patient's record without permission, which is clearly an 'anomalous' event. A degree of subjectivity clearly exists against each of these scenarios. This highlights the fact that the ML detection will only provide part of the story. It cannot and should not be used to make any kind of automated decision that affects a person. This statement aligns with the automated decision guidance provided by the UK Data Protection Act [48]. If ML were to be considered within a Hospital context, any decision process involved needs to be

fully transparent to those affected. Additionally, a policy on what constitutes normal and anomalous behaviours will need to be drawn up by the hospital and publicised to all system users. One final point is that FHIR is a difficult data format to query and manipulate. A recent development is that Google has released a number of FHIR protocol buffers [49] that aim to make working with this healthcare standard more straightforward. Certainly, the creation of FHIR data for this study required considerable effort.

8. Results and Analysis - Availability Scenario

Nine test runs were performed to evaluate and tune the availability detection mechanism using the data generated according to the message ratio described in Sections 4 and 5. The results are presented in Table 8. The table represents a refinement of the algorithm parameters over a number of runs, with A8 and A9 showing final tuning.

Table 8: Availability Scenario Test Results

	Detection Configuration Input		Test Output	
Test	Threshold	Limit Clause Value	Anomalies Detected	Max. msgs/s
A1	1.00	N/A	1	537
A2	1.00	N/A	14	554
A3	1.00	5	17	567
A4	2.00	5	3	567
A5	1.50	5	13	544
A6	1.50	10	8	633
A7	2.00	10	2	633
A8	2.00	10	1	624
A9	2.00	10	1	599

8.1. Availability Scenario Results

The first test (A1) is a Smoke Test with no message flow. The test starts after all data has already been through Iguana and persisted to the SQLite database i.e. no messages are actively flowing through the system. The application's initial query retrieves messages in intervals of 10 seconds. All observations reported a score of less than 1.0, with the exception of a single observation at 22:50:00, which reported a score of 1.2430. This specific observation, and score, correlates with the level shift event. Given that all observations are <1.0 with the exception of the level shift anomaly, the value of 1.0 is used as a starting threshold value for further test runs. Note that when a threshold of 1.0 was set and the test was re-run, a single anomaly was detected, as shown in Figure 12. This demonstrates correct operation of the algorithm.

RUN 1 - Static detection - Combined Dataset A31 Messages - EMA Threshold 1.0 - Interval 10

interval	count
22:47:20	277
22:47:30	926
22:47:40	835
22:47:50	706
22:48:00	627
22:48:10	616
22:48:20	552
22:48:30	560
22:48:40	449
22:48:50	218
22:49:00	918
22:49:10	847
22:49:20	912
22:49:30	1178
22:49:40	1942
22:49:50	1888
22:50:00	**2738**
22:50:10	2640
22:50:20	2919
22:50:30	2538
22:50:40	3216
22:50:50	3161
22:51:00	2424
Total	33087

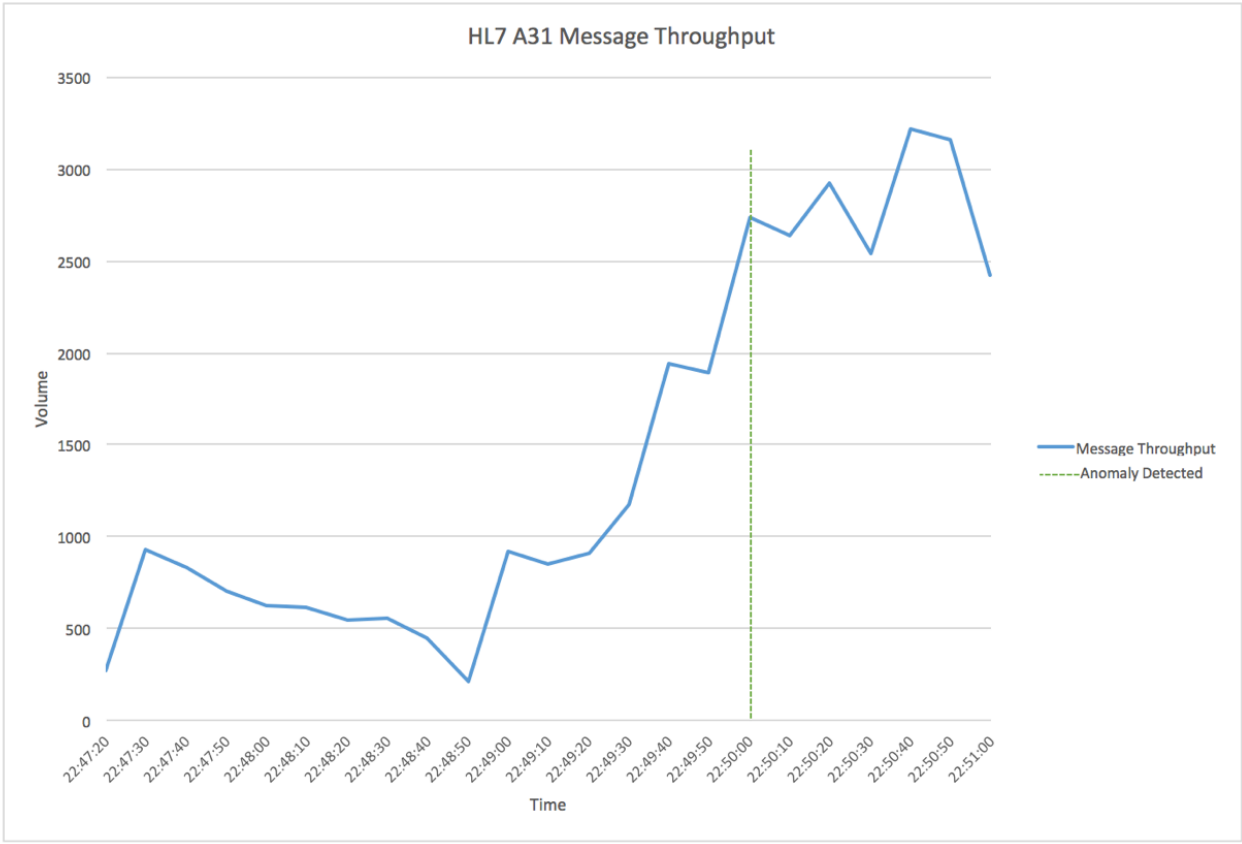

Figure 12: Availability Detection Test A1: Smoke Test

The second test represents a scheduled execution with active message flow i.e. the test simulates data flowing from a PAS, through the integration engine, into the EMR. The application was run with a schedule of 10 seconds and a threshold of 1.0 against messages that are actively flowing into the system. It was observed that each time the detection application runs, all rows in the database are returned for consideration as anomalies. The cumulative inclusion of all rows has the effect of applying the EMA across everything that has been stored to date. This results in a high number of anomalies being detected as the EMA window is considering 'everything' that has been persisted up to the point in time when it runs, every single time. Figure 13 illustrates the volume of anomalies that have been detected. Based on this finding, a 'limit' clause is introduced to the detection application in order to only retrieve the last 'n' records.

RUN 2 - Running detection - Combined Dataset A31 Messages - EMA Threshold 1.0 - Interval 10

interval	count
23:12:30	901
23:12:40	934
23:12:50	903
23:13:00	782
23:13:10	856
23:13:20	875
23:13:30	823
23:13:40	927
23:13:50	871
23:14:00	865
23:14:10	516
23:14:20	799
23:14:30	1453
23:14:40	1825
23:14:50	2132
23:15:00	2766
23:15:10	3255
23:15:20	3212
23:15:30	3030
23:15:40	3123
23:15:50	1393
23:16:00	846
Total	33087

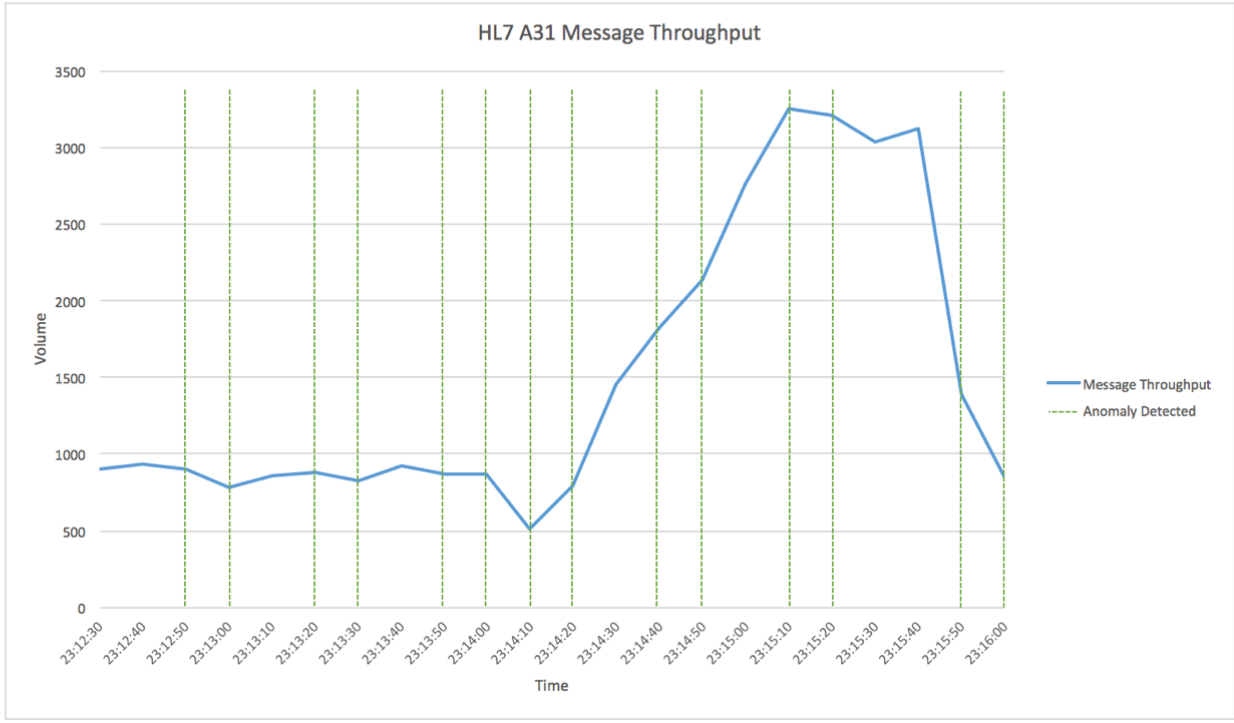

Figure 13: Availability Detection Test A2: High FN Rate

A series of tests were then run varying the limit clause value (i.e. the number of previous observations to include in the algorithm) and the threshold values (i.e. variation in message volume that triggers an anomaly). Note that the query interval is 10 seconds in all tests and the speed of each detection run was approximately 1 second.

For Test A4, 17 anomalies were detected. This is due to the EMA algorithm applying the threshold against a window of 5 observations and the detector continuing to operate even after a level shift anomaly has occurred. The expected behaviour would be to notify an administrator to resolve the anomaly once the level shift is detected and stop further processing.

In Test A5, no level shift was detected. For this test, the threshold was adjusted to 1.5. Twelve anomalies were detected, which is very high. The reasoning for this is that the EMA window of observations (set at 5) is too small, and the threshold of 1.5 is still being reached more easily within the smaller set of observations.

Extending the query limit to pull back the last 10 intervals (with threshold at 1.5 and interval at 10) reduced the number of anomalies detected to 8, due to the greater range of data included, as shown in the Test A6 results. For Test A7, the threshold was raised to 2.0 with the limit maintained at 10. This test proved to be a good indicator of the application working correctly. Two anomalies were detected; the first of which is the level shift.

Finally, for Test A8, additional logic was added to the detection application to terminate on initial detection of an anomaly. With this logic, the detection was stopped after the level shift event (13:02:40). A single anomaly was detected, reflecting a successful run, as shown in Figure 14.

RUN 8 - Running detection - Combined Dataset A31 Messages - EMA Threshold 2.0 - Interval 10 - Limit 10 - Stop after 1st Anomaly Detected

interval	count
12:59:50	233
13:00:00	639
13:00:10	747
13:00:20	654
13:00:30	693
13:00:40	320
13:00:50	580
13:01:00	815
13:01:10	759
13:01:20	455
13:01:30	739
13:01:40	227
13:01:50	260
13:02:00	370
13:02:10	686
13:02:20	365
13:02:30	553
13:02:40	**1119**
13:02:50	723
13:03:00	1640
13:03:10	2553
13:03:20	3161
13:03:30	2850
13:03:40	1621
13:03:50	804
13:04:00	1026
13:04:10	3029
13:04:20	2610
13:04:30	2754
13:04:40	102
Total	33087

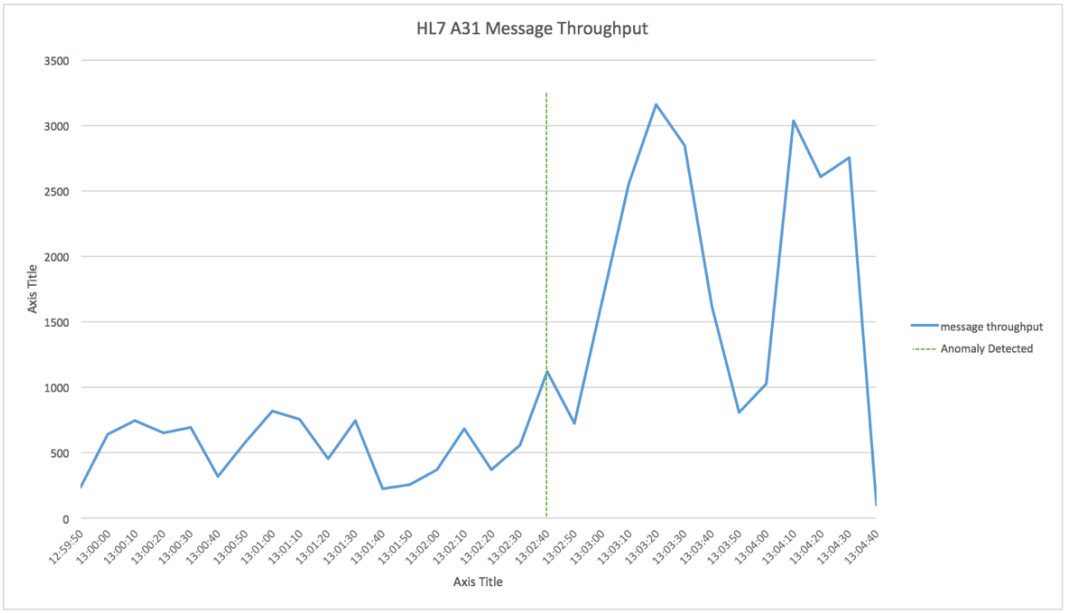

Figure 14: Successful Time Series Anomaly Detection

To confirm the correct operation of the TS anomaly detection application, a final test was conducted against a 'normal' dataset, which contained no level shift events. As expected, no anomalies were detected by the prototype.

The performance measurements for the availability scenario test runs are illustrated in Figure 15. The parameters of accuracy, precision and recall are determined as follows: The detection mechanism aggregates observations into intervals. Each interval can therefore be assessed as TP, FP, FN or TN when measured. The total number of intervals is the total number of items being measured. Only a single True Positive measurement will exist in the Test dataset, which represents the level shift event. Anomalies that are detected before or after the level shift are classified as False Positives.

	TP	FP	TN	FN	Accuracy (TP + TN) / (TP + TN + FP + FN)	Precison TP / (TP+FP)	Recall TP / (TP+FN)
TEST A1	1	0	23	0	**1.00**	1.00	1.00
TEST A2	1	14	22	0	0.62	0.07	1.00
TEST A3	1	17	22	0	0.58	0.06	1.00
TEST A4	1	3	25	0	0.90	0.25	1.00
TEST A5	1	13	21	0	0.63	0.07	1.00
TEST A6	1	8	21	0	0.73	0.11	1.00
TEST A7	1	2	24	0	0.93	0.33	1.00
TEST A8	1	0	23	0	**1.00**	1.00	1.00
TEST A9	0	0	8	0	**1.00**	-	-

Figure 15: Accuracy, Precision and Recall of Availability Tests

- Test A1 was the initial smoke test run, where a single anomaly was detected as expected (after a Threshold of 1.0 was set)
- Test A8 was the successful run, where a single anomaly was detected that correlated with the level shift.
- Test A9 was a test run with no anomalies present in the test dataset.

8.2. Availability Scenario Peformance

The larger the interval size, the greater the number of messages that could pass through the integration engine into the EMR. In Test A8, the message throughput was measured at approximately 624 messages per second. Given a

scheduled run of every 10 seconds, this equates to 6240 messages being processed prior to detection (on a development machine Macbook Pro 2015 16 GB RAM).

It is likely that production throughput would be slower, as each HL7 message has an effect within the downstream EMR system, such as creating a patient or setting up a file plan within the EMR.

From a production perspective, a SQLite database would not be sufficient to persist any great volume of data siphoned from the integration engine and this should be replaced with a more suitable database.

A recommendation from this study would be to model the flow of live HL7 messages within a hospital over an extended period of weeks or months. This would provide a view of the normal operating levels of HL7 message throughput. Synthetic HL7 messages can then be modelled to replicate these usage patterns, and then subsequently be used within a test system to tune the interval, threshold and limit variables.

8.3. Availability Scenario Analysis

A number of observations are apparent when applying TS anomaly detection to identify an availability incident against Evolve EMR. The prototype has successfully detected a level shift during the test runs using the Luminol EMA library. Detection occurs within 10 seconds of the event, which is a significant improvement over the existing manual method. Over the series of 9 test runs, the threshold, limit and interval variables were adjusted in order to obtain a result of 100% accuracy, precision and recall. This is a very positive result given the circumstances of testing.

However, careful consideration needed to be applied when tuning the interval size, limit and threshold variables. The larger the interval size, the greater the number of messages that could pass through the integration engine into the EMR. However, the detection of an availability incident within a 10 second window is a positive result, compared to the current approach which enables messages to flow indiscriminately until manual diagnoses and intervention occurs. This result provides the building block to add a new defensive layer of

security to an EMR system; one that identifies a message surge in real time, yet provides the means to either throttle the system or terminate message acceptance, automatically.

At present, the detection mechanism simply detects. A further extension of the application would be to either throttle the messages (if they are valid) or terminate the channel in Iguana. Within the current system, throttling can be applied for a previously planned and scheduled activity. This does not protect against an availability incident, which is unplanned and unscheduled or deliberately carried out to cause damage. A potential future solution would be to have the detection application set a flag in the database on detection of a level shift. The flag could be checked at regular intervals. If the flag is set, then throttling could be automatically enabled, or the channel could be terminated.

9. Discussion and Future Work

The overall aim of this research is to understand cyber incident detection feasibility against each of the three information security attributes, Confidentiality, Integrity and Availability. The motivation for this work is to protect patient information and ensure that EMR systems are available for use. A failure on either of these elements can have grave implications for the people being treated and the practitioners using the system.

The Confidentiality scenario set out to determine if ML Classification algorithms could detect anomalies in the access patterns of clinicians who view patient records in Evolve IC. The Availability scenario set out to determine if TS anomaly detection algorithms could identify when an unexpected surge of HL7 messages was occurring. This message surge has a detrimental impact on downstream EMR systems.

From the test results obtained from both prototypes (Sections 7 and 8), it can be strongly inferred that the objectives of both prototypes have been met; confidentiality and availability incidents have both been successfully detected. Furthermore, this study has not just focused on the possibility of cyber incident

detection, but also the feasibility, practicality and performance of applying these techniques in a clinical production setting.

The Confidentiality Scenario has exposed some challenges around Data, Tooling and Skill-sets needed for ML adoption. The four variables (appointment, encounter, observation, and break-glass) generate 16 possible combinations of anomaly/ normal events, which in a rule-based system, would equate to 16 individual rules being required. Each additional variable potentially raises the number of rules required by a factor of two.

A rule-based system would quickly begin to struggle with the number of permutations needed to cover all eventualities, in combination with the huge volume of data anticipated in a real hospital setting. ML offers alternative options in this space; if sufficient volume of data is readily available and authorised for use by the Hospital Trust in the system, then unsupervised ML algorithms could be deployed to discern behaviours or patterns in the data. Supervised learning is an option when smaller datasets are available and known specific behaviours are sought in detection routines.

However, deploying a ML based solution could be excessive, if the result can be derived via a rule-based approach. Where a complex interplay of variables is involved, ML can provide an advantage over a rule-based solution, as new incident types can be accommodated by adding them to the Training Set. However, if the expectation is that the number of variable combinations and permutations to be observed is small, the expected incidents are low in volume, or the rules are well known, then a traditional SIEM may be a more appropriate solution. This should be considered in conjunction with the financial, time and complexity factors in building a ML system, and associated pipeline. For example, specialist ML knowledge and training is required to train, tune and retrain the model. Even if the case for ML is strong, careful judgement should be applied as to how it is used in a clinical setting. It should not be used as the basis for automatic judgements or decisions. A policy should be in place that clearly communicates expected EMR use to system users.

The Availability Scenario has shown good results. A successful detection

of the availability incident occurs within 10 seconds of the level shift event. However, this took multiple rounds of tuning the threshold, limit and interval variables, with only a single one of these variables being modified each time on each test run. This tuning period must be carried out in order to ensure that the correct values are being identified and used. If this stage is performed incorrectly, then the result will be that too many anomalies will be detected, or alternatively, too many messages will bypass the detection mechanism. Nevertheless, detection within 10 seconds of the event is a promising initial result and is a significant advantage over the existing manual method.

While this research work focused on the EMA algorithm, the ARIMA algorithm also shows promise for both incident detection and prevention. ARIMA specifically looks into predicting future states, given previous observations, whereas the EMA approach used in the Availability detector operates on previously seen data i.e. an incident will already have happened when it is detected. ARIMA could offer the means to predict a potential incident and prevent its negative impact. This will be explored in future work.

A further extension to the availability scenario will be extension of the solution to automatically enable throttling within the integration engine as a defensive mechanism to protect the downstream EMR system.

Finally, with respect to data, due to the limitation on access to live clinical data, synthetic data was generated for this research. In a production system, several aspects must be considered for the protection of data. The handling of data in acccordance with Data Protection, GDPR and HIPAA regulations must be ensured. In addition, security controls must be applied to the data stores and pipeline system of the proposed cyber incident detection applications including ML and TS test data repositories.

10. Conclusion

This research has demonstrated a practical application of cyber incident detection against real EMR systems. Through this work, the feasibility of Ma-

chine Learning and Time Series based solutions have been demonstrated whilst highlighting the wider implications which need to be considered if deploying a ML solution in a clinical setting. For the confidentiality scenario, the Support Vector Machine produced the best results with 98.94% accuracy. SVM also achieved a figure of 100% accuracy when exposed to previously unseen data indicating robustness of the algorithm. The Availability prototype successfully demonstrated detection of a HL7 message surge, an event which caused a serious negative impact on a U.K. hospital in September 2016. To the best of our knowledge, this work provides the first development and analysis of a ML-based and time series anomaly detection solution for cyber incident detection for EMR systems.

References

[1] K. Sheridan, "Major Cyberattacks On Healthcare Grew 63% In 2016", Dark Reading, http://www.darkreading.com/attacks-breaches/major-cyberattacks-onhealthcare-grew-63--in-2016/d/d-id/1327779 (2016).

[2] "Rise in hospital cyber attack reports", BBC News, http://www.bbc.com/news/uk-england-oxfordshire-39556062 (2017).

[3] C. Humer, J. Finkle, "Your medical record is worth more to hackers than your credit card", Reuters, http://www.reuters.com/article/us-cybersecurityhospitals-idUSKCN0HJ21I20140924 (2014).

[4] "Fifth Annual Study on Medical Identity Theft", http://medidfraud.org/wpcontent/uploads/2015/02/2014_Medical_ID_Theft_Study1.pdf (2015).

[5] W. Zhang, X. He, An anomaly detection method for medicare fraud detection, in: Big Knowledge (ICBK), 2017 IEEE International Conference on, IEEE, 2017, pp. 309–314.

[6] A. Patterson, "Healthcare Industry Wisdom on Medical Identity Fraud", http://medidfraud.org/paper-healthcare-industry-wisdom-onmedical-identity-fraud/ (2016).

[7] A. Patterson, "PHI: Valuable and Vulnerable", http://www.fortherecordmag.com/archives/0316p18.shtml (2016).

[8] "DDoS Case Study: DDoS Attack Mitigation Boston Childrens Hospital", Radware, https://security.radware.com/ddos-expertsinsider/ert-case-studies/boston-childrens-hospital-ddos-mitigation-case-study/ (2015).

[9] "Doctor Who Stole Personal Information of Nearly 100,000 NRAD Associates Patients Arrested", https://www.databreaches.net/doctorwho-stole-personal-information-of-nearly-100000-nrad-associates-patients-arrested-asresult-of-joint-ncda-nc/ (2017).

[10] D. McGlade, "daveym/msc", http://github.com/daveym/msc (2017).

[11] F. Rezaeibagha, K. T. Win, W. Susilo, A systematic literature review on security and privacy of electronic health record systems: technical perspectives, Health Information Management Journal 44 (3) (2015) 23–38.

[12] J. L. Fernández-Alemán, I. C. Señor, P. Á. O. Lozoya, A. Toval, Security and privacy in electronic health records: A systematic literature review, Journal of biomedical informatics 46 (3) (2013) 541–562.

[13] P. E. Idoga, M. Agoyi, E. Y. Coker-Farrell, O. L. Ekeoma, Review of security issues in e-healthcare and solutions, in: HONET-ICT, 2016, IEEE, 2016, pp. 118–121.

[14] "Securing Hospitals: a research study and blueprint", https://securityevaluators.com/hospitalhack/securing_hospitals.pdf (2016).

[15] O. Boric-Lubecke, X. Gao, E. Yavari, M. Baboli, A. Singh, V. M. Lubecke, E-healthcare: Remote monitoring, privacy, and security, in: Microwave Symposium (IMS), 2014 IEEE MTT-S International, IEEE, 2014, pp. 1–3.

[16] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, Y. Xiang, Detecting and preventing cyber insider threats: A survey, IEEE Communications Surveys & Tutorials.

[17] X. Li, Y. Xue, Y. Chen, B. Malin, Context-aware anomaly detection for electronic medical record systems, in: USENIX Workshop on Health Security and Privacy, Vol. 2011, NIH Public Access, 2011, p. 8.

[18] Y. Chen, W. Xie, C. A. Gunter, D. Liebovitz, S. Mehrotra, H. Zhang, B. Malin, Inferring clinical workflow efficiency via electronic medical record utilization, in: AMIA Annual Symposium Proceedings, Vol. 2015, American Medical Informatics Association, 2015, p. 416.

[19] A. Mohan, A medical domain collaborative anomaly detection framework for identifying medical identity theft, in: Collaboration Technologies and Systems (CTS), 2014 International Conference on, IEEE, 2014, pp. 428–435.

[20] C. Di Sarno, V. Formicola, M. Sicuranza, G. Paragliola, Addressing security issues of electronic health record systems through enhanced siem technology, in: Availability, Reliability and Security (ARES), 2013 Eighth International Conference on, IEEE, 2013, pp. 646–653.

[21] Y. Chen, S. Nyemba, B. Malin, Detecting anomalous insiders in collaborative information systems, IEEE transactions on dependable and secure computing 9 (3) (2012) 332–344.

[22] F. Siemons, "Developments in Machine Learning vs. Traditional SIEM Solutions", `http://resources.infosecinstitute.com/developments-inmachine-learning-vs-traditional-siem-solutions/` (2016).

[23] "FHIR Release 3 (STU)", https://www.hl7.org/fhir/ (2017).

[24] "Evolve Electronic Medical Record", Kainos Evolve, https://www.kainosevolve.com/electronic-medical-record/ (2017).

[25] "Evolve Integrated Care", Kainos Evolve, https://www.kainosevolve.com/integrated_care/ (2017).

[26] "HL7 Standards Product Brief - HL7 Version 2 Product Suite", http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185 (2017).

[27] "JSON", http://www.json.org/ (2017).

[28] C. Chio, D. Freeman, Machine Learning and Security, O'Reilly Media, 2018.

[29] M. Ahmed, A. N. Mahmood, J. Hu, A survey of network anomaly detection techniques, Journal of Network and Computer Applications 60 (2016) 19–31.

[30] E. Denham, "Four lessons NHS Trusts can learn from the Royal Free case", https://iconewsblog.org.uk/2017/07/03/four-lessons-nhs-trusts-canlearn-from-the-royal-free-case/ (2017).

[31] Y. Pan, F. Sun, J. White, D. C. Schmidt, J. Staples, L. Krause, Detecting web attacks with end-to-end deep learning.

[32] "AutoEncoders are essential in deep neural nets - towards data science", Towards Data Science, https://towardsdatascience.com/autoencoders-are-essential-in-deep-neural-nets-f0365b2d1d7c (2018).

[33] N. Moradpoor, M. Brown, G. Russell, Insider threat detection using principal component analysis and self-organising map, in: Proceedings of the 10th International Conference on Security of Information and Networks, ACM, 2017, pp. 274–279.

[34] "Guide to the General Data Protection Regulation (GDPR)", Ico.org.uk, https://ico.org.uk/for-organisations/guide-to-the-general-data-protection-regulation-gdpr/ (2018).

[35] "Incident Management", NCSC, https://www.ncsc.gov.uk/incident-management (2017).

[36] "Confidentiality, Integrity, and Availability", Mozilla Developer Network, https://developer.mozilla.org/en-US/docs/Web/Security/Information_Security_Basics/Confidentiality,_Integrity,_and_Availability (2016).

[37] "Wannacry Ransomware: What it is and how to protect yourself", http://www.wired.co.uk/article/wannacry-ransomware-virus-patch (2017).

[38] A. L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Communications Surveys & Tutorials 18 (2) (2016) 1153–1176.

[39] "Anomaly Detection of Time Series Data Using Machine Learning & Deep Learning", Xenonstack, https://www.xenonstack.com/blog/anomaly-detectionof-time-series-data-using-machine-learning-deep-learning (2017).

[40] "linkedin/luminol", https://github.com/linkedin/luminol (2017).

[41] "synthetichealth/synthea", The MITRE Corporation, GitHub, https://github.com/synthetichealth/synthea (2016).

[42] "Explore Synthetic Mass", Syntheticmass.mitre.org, https://syntheticmass.mitre.org/dashboard/index.html (2016).

[43] "UsingPickle - Python Wiki", https://wiki.python.org/moin/UsingPickle (2017).

[44] D. Bader, "dbader/schedule", http://github.com/dbader/schedule (2017).

[45] "3.1. Cross-validation: evaluating estimator performance scikit-learn 0.19.0 documentation", http://scikitlearn.org/stable/modules/cross_validation.html (2017).

[46] J. Brownlee, "How to Evaluate Machine Learning Algorithms - Machine Learning Mastery", http://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/ (2017).

[47] "scikit-learn: machine learning in Python scikit-learn 0.19.0 documentation", http://scikitlearn.org/stable/ (2017).

[48] "Automated decision taking", https://ico.org.uk/fororganisations/guide-to-data-protection/principle-6-rights/automated-decision-taking/ (2017).

[49] "Making Healthcare Data Work Better with Machine Learning", Research blog, https://research.googleblog.com/2018/03/making-healthcare-data-work-better-with.html (2018).