Improving the Prediction of Radiation Therapy Patient Waiting Times Through Visualizations, Comparison of Algorithms, and Data Analysis

COMP 396 Undergraduate Research Project

Maxim Gorshkov[1]
Supervisor: Professor Laurie Hendren[1]

[1] School of Computer Science, Faculty of Science, McGill University

Abstract

In radiation oncology, patients experience a certain waiting time between their initial CT-Scan and the start of their treatment. Although highly structured and procedural, because of the number of departments and people involved in the 8 steps between the two, the waiting time can vary from a few days to many weeks. In this paper, we use a tool set of unique and exhaustive data filtering, distinct data visualizations, and a survey of algorithms expand upon the accuracy and efficacy of waiting time prediction. Patient records from the Montreal General Hospital are used with a combination of modern computing and frameworks to present and explore the data. With 9 visualizations presented in this paper, both patients and doctors are targeted. With the patient-oriented graphs, a more clear breakdown of each stage shows the context of the stage within the total waiting time. The doctor-driven charts help to identify problems with data and to increase efficiency. As a result of the filtering, fewer assumptions needed to be made with the data, resulting in more exhaustive and comprehensive data used in the training for the 5 predictions algorithms compared. Finally, a Gaussian Process algorithm gave a predictive waiting time accurate to 9.08 days. By combining results in this paper with more data and specific factors in the data trends, future would could lead towards a better estimate. Using other combinations of visualizations, future work also lead towards substantially more uniform treatment by different oncologists.

1 Introduction

When a patient is referred to a radiation oncologist, and a treatment is required, a CT-Scan is taken. Between this CT-Scan and the start of treatment, there is an associated, variable, waiting time. Although there are 8 distinct stages of the waiting time, as explored in this paper, the current estimate for the patient is set to ”about two weeks” for the whole process.

The goal of this paper is to expand on the ground breaking work done to establish certain guidelines and collect data for an analysis of the factors involved in predicting waiting times [1-2]. Primarily, by using a combination of data processing and more intuitive data representation, more accurate data can be obtained to work with an estimation algorithm for patient waiting times. In turn, this would help to decrease the anxiety of the patients, by giving access to clear information regarding treatment and associated waiting times. This would also help keep the staff informed on the latest data regarding current patients and overall waiting time statistics to work towards increasing efficiency.

This report will begin with an outline of the expected patient waiting times in radiation oncology. Afterwards, the data-set from the Montreal General Hospital (McGill University Health Centre) on patient radiotherapy will be analyzed in several forms to aide in prediction of waiting times. The mapping of the database relational model, data sanitizing, and visualizations are discussed. Afterwards five predictive algorithms are compared.

2 Background

2.1 Patient Waiting Times

A typical patient in the Radiation Oncology department has to go through several steps between the initial consultation from the patient's primary physician through to the start of the treatment. For the purpose of the paper, the eight following steps are used for analysis: *CT-Scan, Initial Contour, MD Contour, CT Planning Sheet, Dose Calculation, MD Approve, Physics QA, Ready for Treatment*. The full treatment planning course is shown in Figure 1. At each of the eight steps in the treatment planning process, there is an associated waiting time for the patient as the patient data gets passed between different clinicians, medical physicists, and administrative offices. Depending on the type and severity of the diagnosis, each of these steps may fluctuate. As such, there is a standard practice to tell a patient that the full treatment planning process will take "about two weeks". In practice, however, this can be anywhere between a few days and many weeks.

For the purposes of all waiting times in this paper, only the time point at which the appointment or task begins is considered. Thus, the waiting time for the patient is between the start of any two events in the flow described in Figure 1. The total time the patient has to wait for the planning procedure, is there from the start of the *CT-Scan* to the start of the *Ready for Treatment* phases.

2.2 Generated Data Set

The generated data set containing the patient information, as well as the waiting times and scheduling details associated with each of the planning stages takes the form of a relational database. Each patient receives a serial number and each part of the planning stage is associated with the patient using this key. Although the original database uses a complex schema, a subset of this data was used to perform all analysis in this paper [1]. Mainly, only tables with concrete waiting times for one of the eight stages being studied, doctor information, as well as details of the specific cancers were extracted.

Note that all personal data pertaining to patients and doctors was removed from the subset used in the analysis. Also note that the patient data is valid between January 2012 and February 2015.

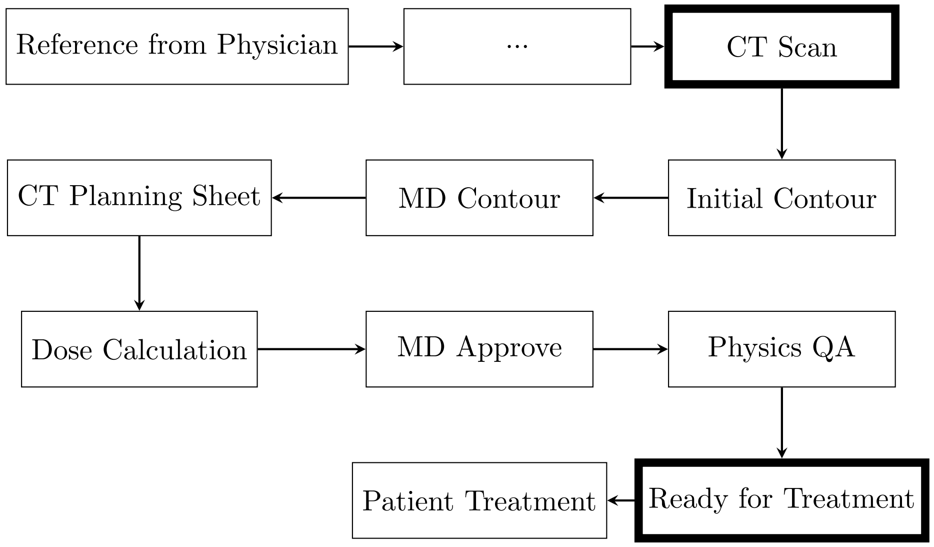


Figure 1: Truncated Radiation Oncology Planning Process. Full diagram can be found in a paper by Leung [1]. Primary focus of this paper is on the stages between the two bolded boxes.

3 Related Work

Patient waiting times, and the analysis of the possible reduction has had a lot of attention both in the field of Oncology and other fields in medicine. Although we are focusing more on the prediction of waiting times associated with the planning process, it is important to note that options are being explored to standardize and reduce the associated waiting times. For instance, in a 2006 paper, Danjoux et al. investigate the progress made at a Toronto Radiotherapy unit and concluded that since the initiation of a "Rapid Response" program, nearly 90% of patients were seen within 2 weeks of referral, 85% were simulated on the day of their initial consultation, and 60% percent started their RT treatment on the day of their consultation visit. The overall median interval from referral to treatment was 8 days [3]. The interval of 8 days is already an improvement over the standard "two week" mark that is given to patients at the MGH.

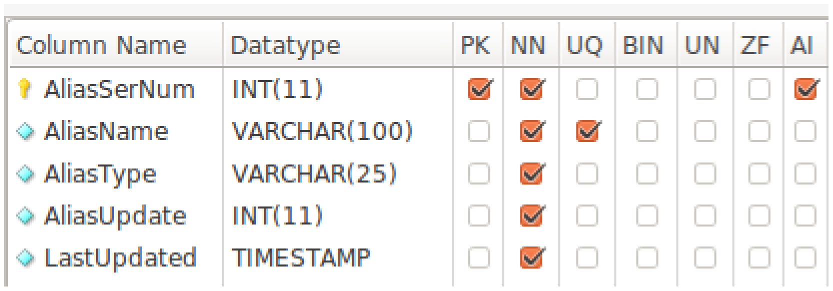
Aside from patient waiting times, machine learning models are used in medicine to classify and diagnose. In a recent paper, WEKA is used to classify Breast Cancer. [4]. Some of the findings here suggest that there are certain algorithms that produce results which classify with high accuracy - 89.71% and are computationally efficient. With the breakdown of these algorithms, we can test to see if waiting times also follow similar trends.

4 Database Object/Relation Mapping

Even with the simplification of the database model through the removal of unnecessary tables, analysis of data on an SQL database is inherently difficult due the limitations of the language. As such, each relevant table was mapped to a sequence of objects. In this case, the Hibernate ORM was used to achieve a graph of objects in Java [5]. Although the use of object/relation mapping does not remove the need for having a relational database, it makes reasoning about data, and performing operations on this data more intuitive and certainly more powerful.

An example of the table "Alias" is presented in Figure 2. The same table, represented as a Java object is shown in Figure 3. Note that it is possible to specify the same level of constraints on the data, and to assign types to each column in the table, which becomes a field in the object. A full description of the requirements for mapping a table is presented in Appendix 1.

Once each of the relevant tables were mapped to its respective object, a "Patient" and "DataPoint" object were created in order to aggregate the data found throughout the set. In the web application designed for this paper, the relevant data is loaded whenever the application starts. All data is assigned to a specific "Patient" object through "DataPoint" objects created for each waiting time that the database contains for this specific patient serial number. Filtering is then performed on this data as described in the next section.



Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
AliasSerNum	INT(11)	✓	✓	☐	☐	☐	☐	✓
AliasName	VARCHAR(100)	☐	✓	✓	☐	☐	☐	☐
AliasType	VARCHAR(25)	☐	✓	☐	☐	☐	☐	☐
AliasUpdate	INT(11)	☐	✓	☐	☐	☐	☐	☐
LastUpdated	TIMESTAMP	☐	✓	☐	☐	☐	☐	☐

Figure 2: The "Alias" table described in MySQL formatting, including the options for each column.

```
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
@Basic(optional = false)
@Column(nullable = false)
private Integer AliasSerNum;
@Size(max = 100)
@Column(length = 100, nullable = false)
private String AliasName;
@Size(max = 25)
@Column(length = 25, nullable = false)
private String AliasType;
@Size(max = 11)
@Column(length = 11, nullable = false)
private Integer AliasUpdate;
@Column(nullable = false)
private Date LastUpdated;
```

Figure 3: The "Alias" object as a Java class, and subsequently as used for this paper.

5 Data Sanitizing and Filtering

At each of the steps in the waiting time, data is recorded either automatically or through manual input. As with most tasks involving manual input, there is a certain factor of human error which leads to discrepancies. In order to provide improvement on the predictions, great care was taken to ensure that each patient considered in the paper had a complete set of data. By performing rigorous filtering on the data, instead of having to make numerous assumptions about what certain data means, the final data fed into the prediction model can be as complete and comprehensive as possible.

The filtering results are shown in Figure 4 and each step is described in detail below. Please note that the following values are instances of the web application run on April 24, 2015.

Filtering Name	**Patients Disqualified**	**Number of Patients**
Initial		35426
Initial Appointment	29859	5567
Naive Stage	256	5311
Advanced Stage	1213	4098
Chronological Stage	790	3308
Remove Outliers	706	2602
Final		2602

Figure 4: Results from the filtering at the start of the web application

5.1 Filter 1: Initial Appointment

While the data is being mapped to "DataPoint" objects, and subsequently to the "Patient" objects, if there is a patient without the initial appointment: a value for the *CT-Scan*, the patient is no longer included in the data set.

5.2 Filter 2: Naive Stage

After all of the waiting times are assigned to specific patients, a simple filter runs through to check the completeness of the stages of the planning process. Since there are eight steps considered, if there are less than eight "DataPoint" objects to a "Patient" object, this patient is removed from the data set.

5.3 Filter 3: Advanced Stage

The advanced stage filter works in a similar fashion to filter 2. Instead of just checking for the number of overall stages recorded for a specific patient, the filter also checks the types of stages recorded. If there are eight unique stages, so, one of each stage, then the patient is included in data set. Otherwise, the patient is excluded.

5.4 Filter 4: Chronological Stage

The final filtering on the stages does an ordering of the stages chronologically. If there is an ordering such that the stages occur one after another in the proper sequence, this ordering is taken as the final data for this patient. If there is no such order, the patient is excluded from the data set.

5.5 Filter 5: Remove Outliers

The last filter on the data set checks the data points for any given stage for trends in data. First off, if there are any waiting time between two stages which is greater than 35 days, the patient associated is removed. Next, if the waiting time between any two stages is more than 3 standard deviations away from the mean, the patient is again removed from the final data set.

6 Visualizations

The visualizations of data were developed with two main benefactors in mind, the patient and the medical professional. From the perspective of the patient, they are able to access the progress of their waiting times so far, and, are able to see the prediction of the waiting times that remain. In the case of the medical professional, they are able to see patient progress for individual patients, patients who have the same diagnosis, and see the trends of waiting times.

6.1 Patient

The waiting time progress of a patient shows the waiting times between any two steps all on one bar chart. This way it is easier to compare how long each step has taken in relation to one another. An example for a patient is shown in Figure 5. Every bar, thus every waiting time can be enabled/disabled to waiting times for subsets. This visualization gives a graphical meaning to waiting times that the patients experience and help to identify for both the patient and the doctor which sections are causing the most delay. Another view for patients is envisioned to be included alongside this progress view with the prediction of the remaining waiting time. This ”Timeline View” is discussed in the next section with the prediction algorithms, as well as in the future works section.

In order to put waiting times into more intuitive terms, the calendar view was created, as shows in Figure 6. This overlays the stages of the waiting times onto a calendar and allows the patient to see the waiting times in terms of days of the week, and also in the context of other plans which have have been made over this time period. Since the data points for each waiting time have a timestamp, the calendar is generated through the framework.

These two views give a relative representation of the waiting time to one another. However, in the future it would work well to remove waiting times which are not caused by the hospital. For example, removing the weekends, holidays, and other hospital closures from the waiting time may give a more accurate estimate of the waiting time due to the planning of the treatment.

6.2 Doctor

The visualizations available for each doctor serve a few purposes. There are specific views which help with data analysis, and others which put into perspective each individual journey of a patient.

The first data analysis view, shown in Figure 6, the calendar view, is described in the previous subsection. Not only is this helpful to visualize the waiting times in terms of concrete days, but it also helps to distinguish waiting times which are a result of processing at the hospital, as opposed to weekends and holidays. The calendar view also breaks the stages down by hour and minute if more than one of these stages occurs on a particular day.

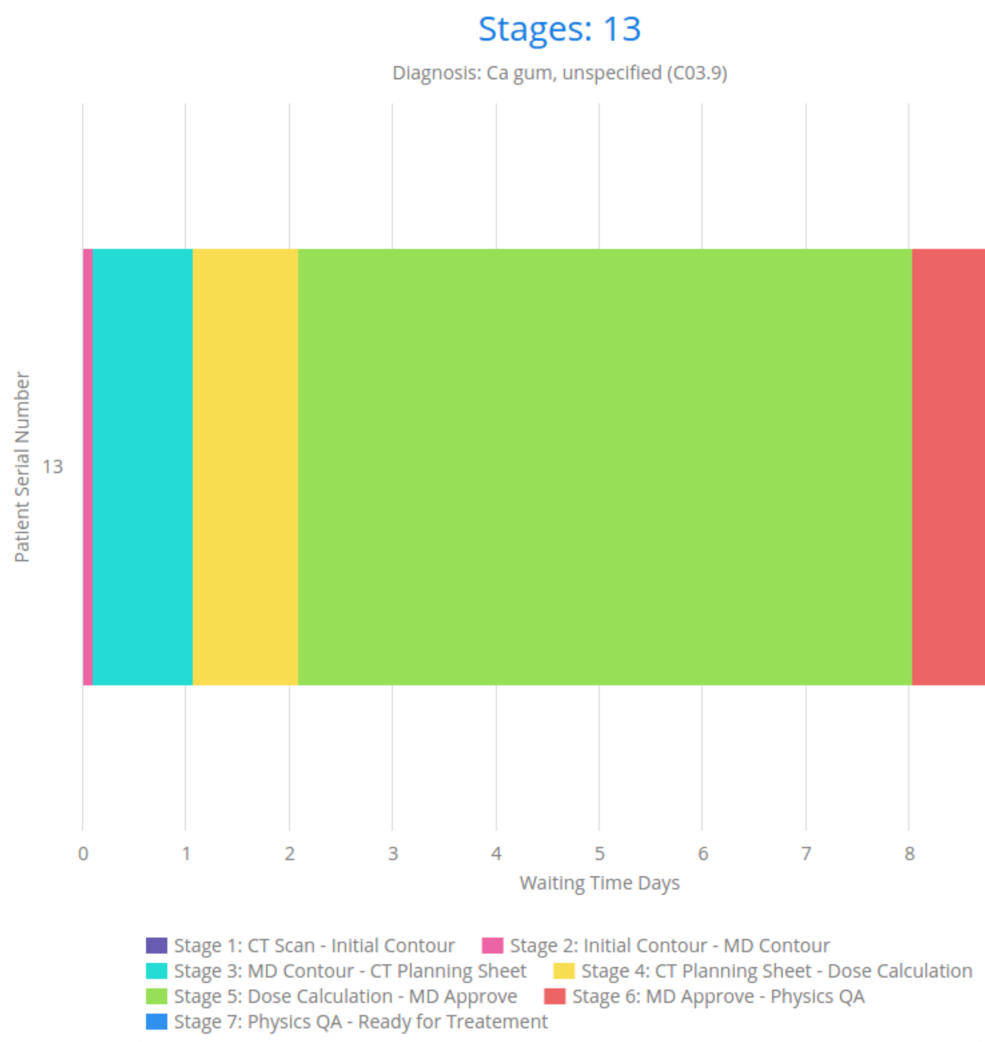


Figure 5: Individual patient waiting times per stage with all 7 waiting times shown. Note that the scale is in days.

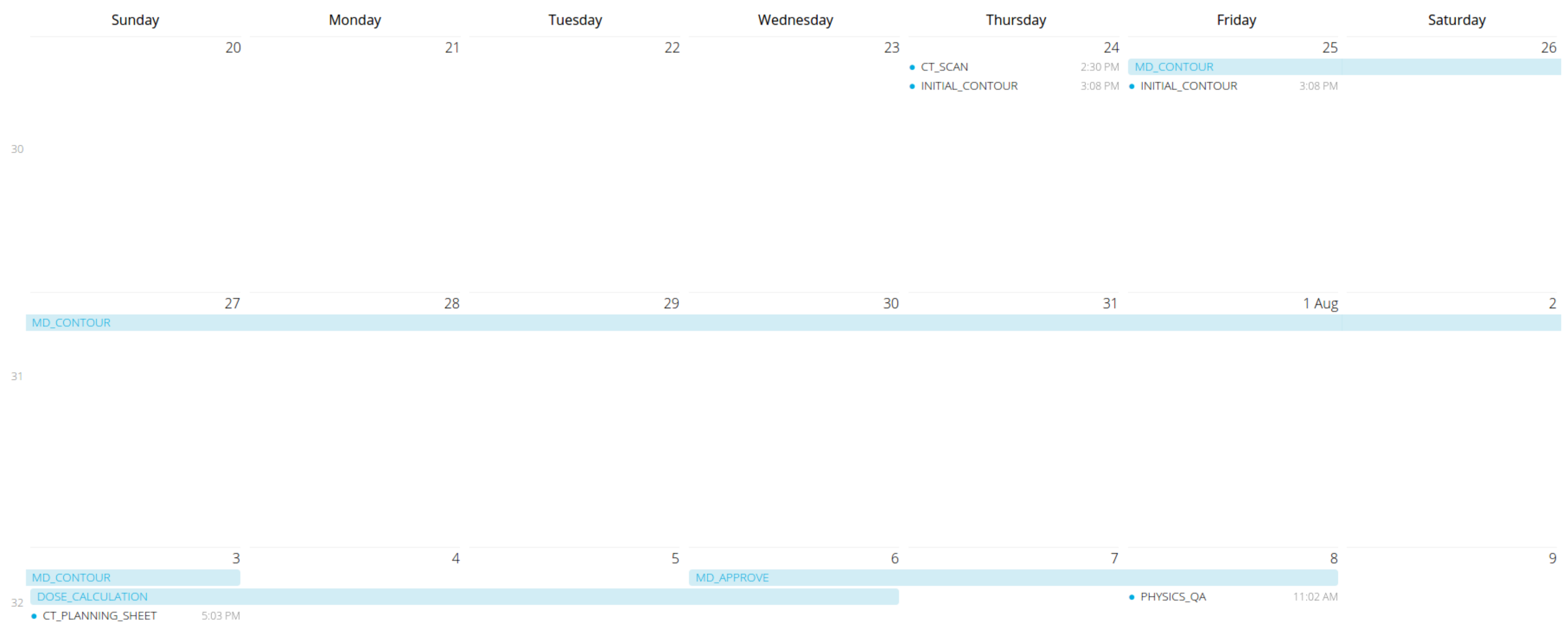


Figure 6: Individual patient waiting times over a calendar. Note that each waiting time appears as an overlay onto the calendar.

The two main views which help to address the individual journeys of patients for a single oncologist are presented in Figure 7 and Figure 8. Figure 7 shows the journey of patients as seen in the previous subsection: as a stacked bar. This way they are able to see each of the steps in the waiting time. This chart is ordered by the total waiting time that a patient experiences. This view is helpful to identify outliers as well as to indicate which common stages between patients have a long waiting time. This view can also be filtered by the diagnosis for further inspection.

Figure 8 presents the same data as Figure 7 but ordered in chronological order. This means that the least recent patient on the left through to the most recent patient on the right. This view helps to identify trends in patient waiting times over the course of an oncologists' career. Subsequently, outliers can be identified in this view, and the diagnosis can be specified in the graph for further inspection.

In both cases, having the ability to reorder the data based on certain parameters, such as in the increasing order of a specific waiting stage, can work towards having a more tailored analysis of data for a specific oncologist. For the second view with chronological order, having some sort of averages for the stages displayed can provide a sense of how usual the waiting times are for a specific patient, in a glance.

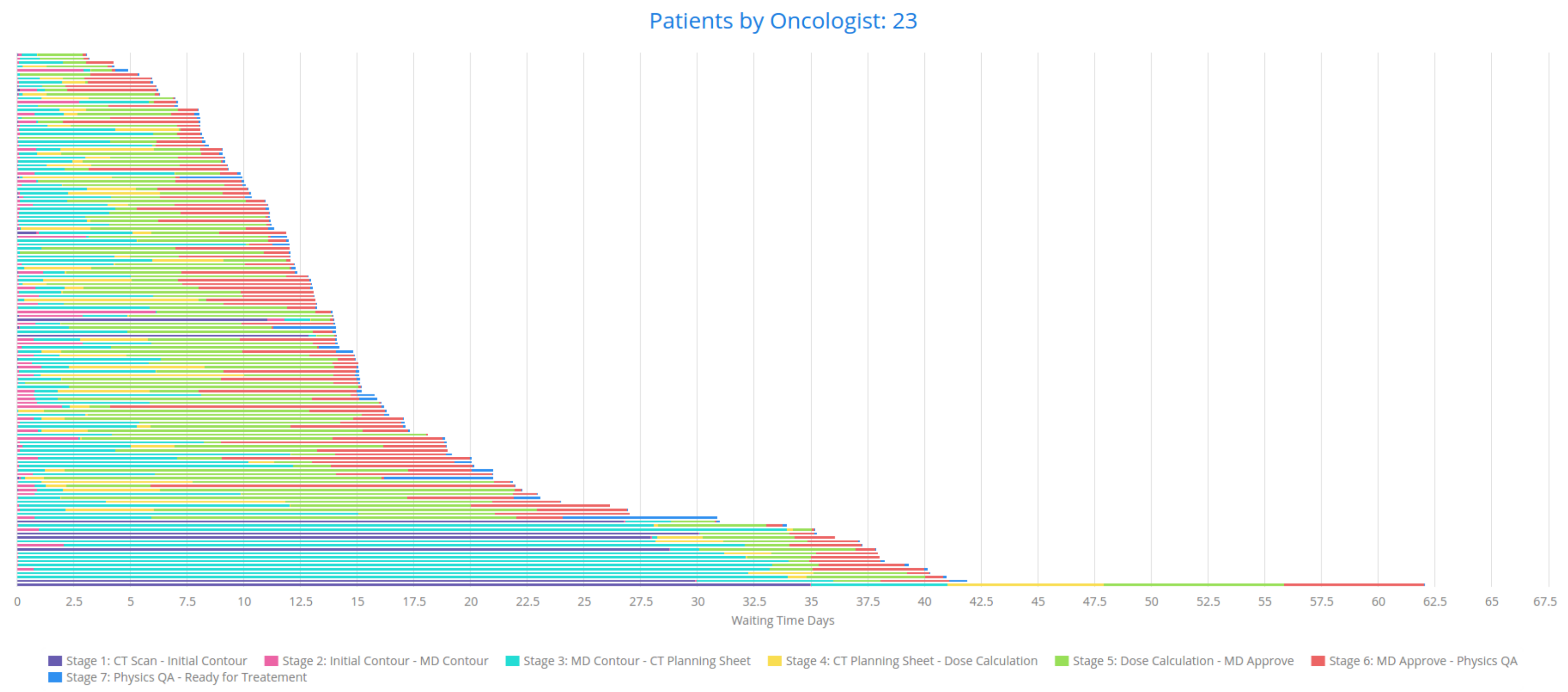


Figure 7: Journey of patient waiting times for a specific oncologist, ordered by the total waiting time.

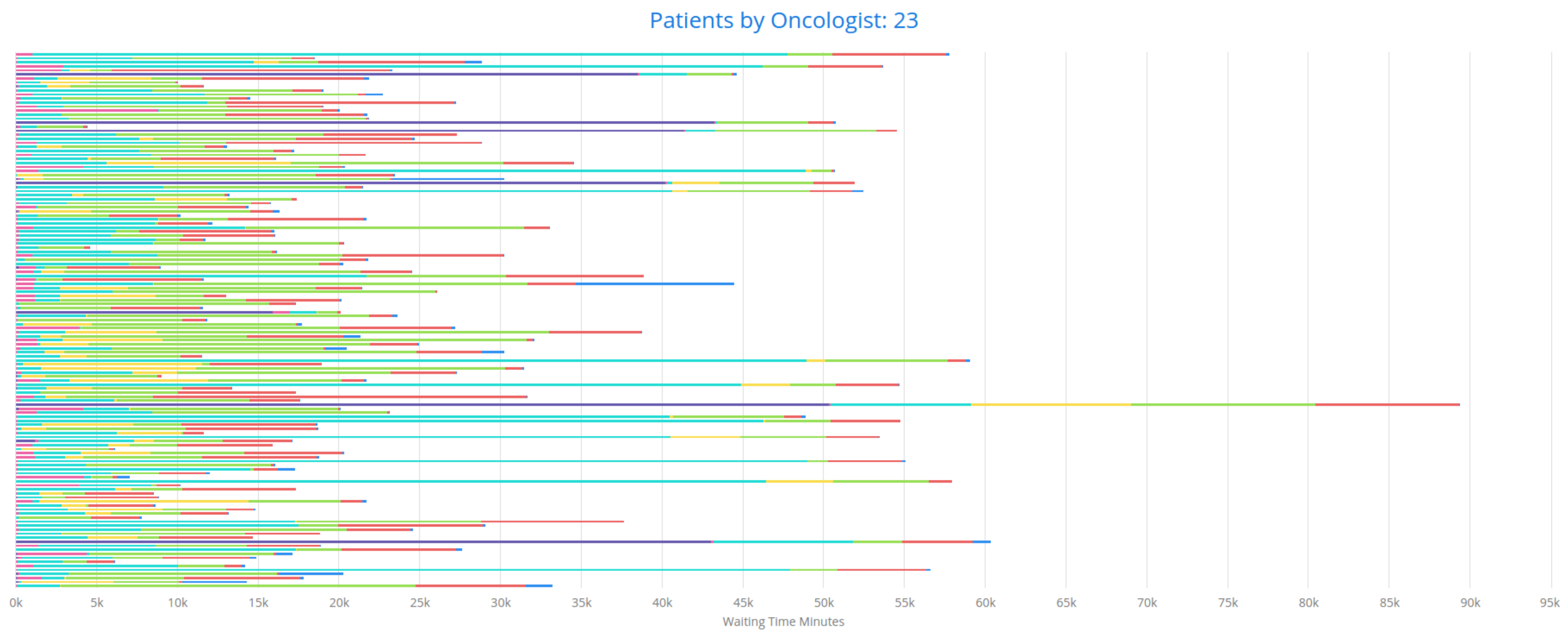


Figure 8: Waiting time for each stage as a function of time. Note that the patients are ordered in chronological order.

7 Analysis of Predictive Algorithms

After the data was filtered as described in a previous section of this paper, the data set could be used to perform predictions on future waiting times. In general, the data was explored and different algorithms were tested using the WEKA Machine Learning library [6]. At first, all of the factors presented by Leung [1] were tested: diagnosis, oncologist, age and priority level. It was noted that having so many exploratory variables decreased the total number of algorithms that could be explored on the data set. As such, for this paper, only the patient's priority level, oncologist, and waiting times by each step of the planning progress are considered. For each of the 5 algorithms, the training set consisted of 4/5 random patients from the data-set and the testing set consisted of 1/5 random patients. Wherever possible, cross-validation of 10 was performed on the testing set. For the testing set, each patient randomly has a cut from a certain step onwards to represent how much waiting time we know from a patient versus how many steps the patient has left to the start of treatment. The accuracy of the algorithm was measured with the average of the absolute value between the prediction and actual result. The accuracy of each algorithm is presented in Figure 9 and the description of the algorithm with the best performance is shown below.

Algorithm	Subtype	Accuracy (Days)
Gaussian Process	RBF Kernal	9.09
Least Med Squares		21.22
Multilayer Perceptron		39.98
M5Rules		30.22
M5P		24.15

Figure 9: Comparison of the accuracy of the algorithms.

The Gaussian Process algorithm is a model of supervised learning in the form of regression. It works in a similar way to a support vector machine but instead does a processing over functions. Thus, this is a useful model to specify very flexible non-linear regressions [7].

The Gaussian Process algorithm was applied to a test set of 21 patients to predict the waiting times given a specified cut off point. This prediction is shown in Figure 10. Everything to the left of the black line is known data, given the stage started and completed. If the stage was in progress, the difference between the starting point and the cut-off is taken. Everything to the right is a prediction of the rest of the waiting stages. The prediction and the actual data is compared in Figure 11.

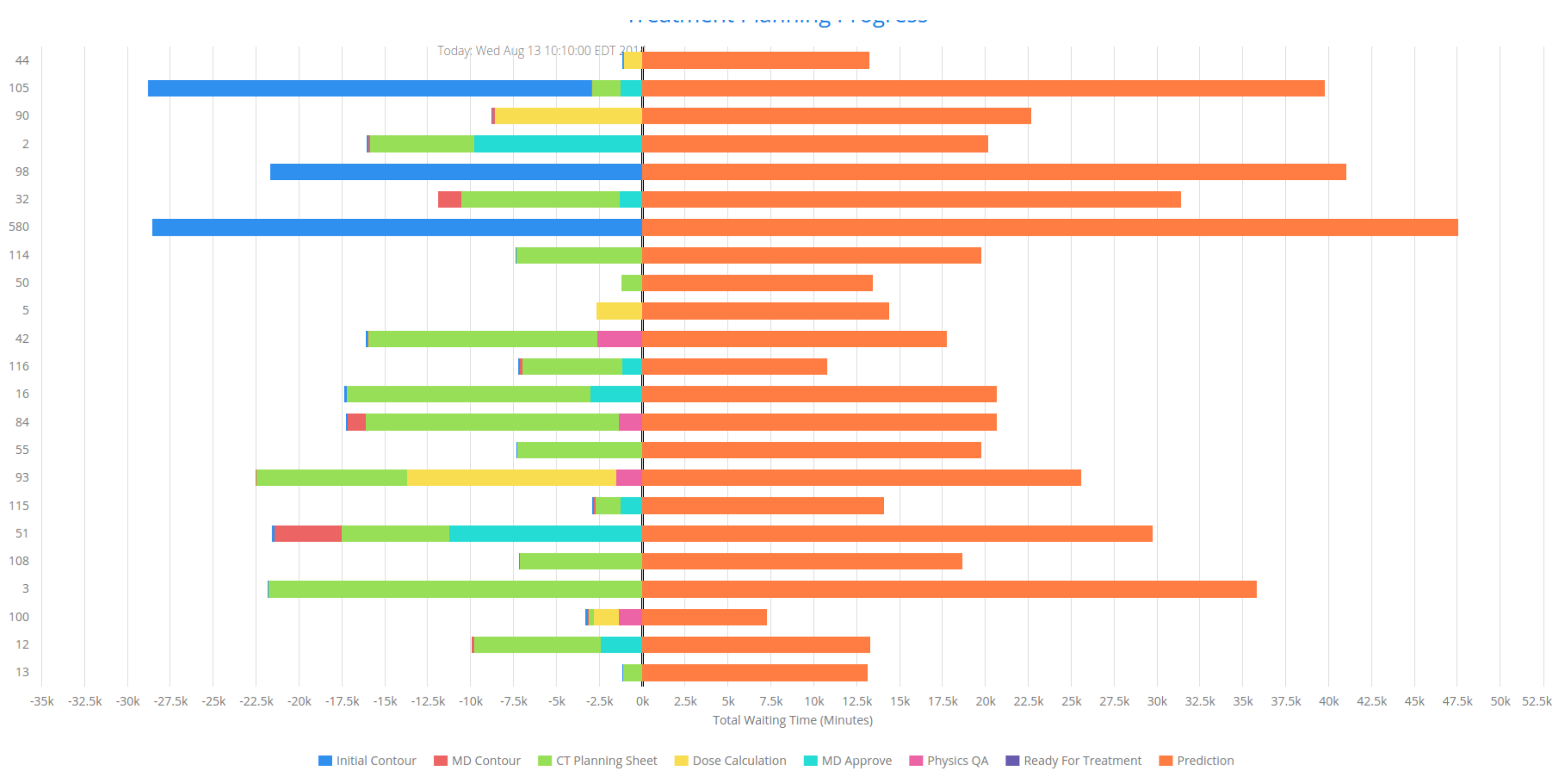


Figure 10: Prediction of waiting times for patients based on current stages completed. Note that everything to the right is predicted.

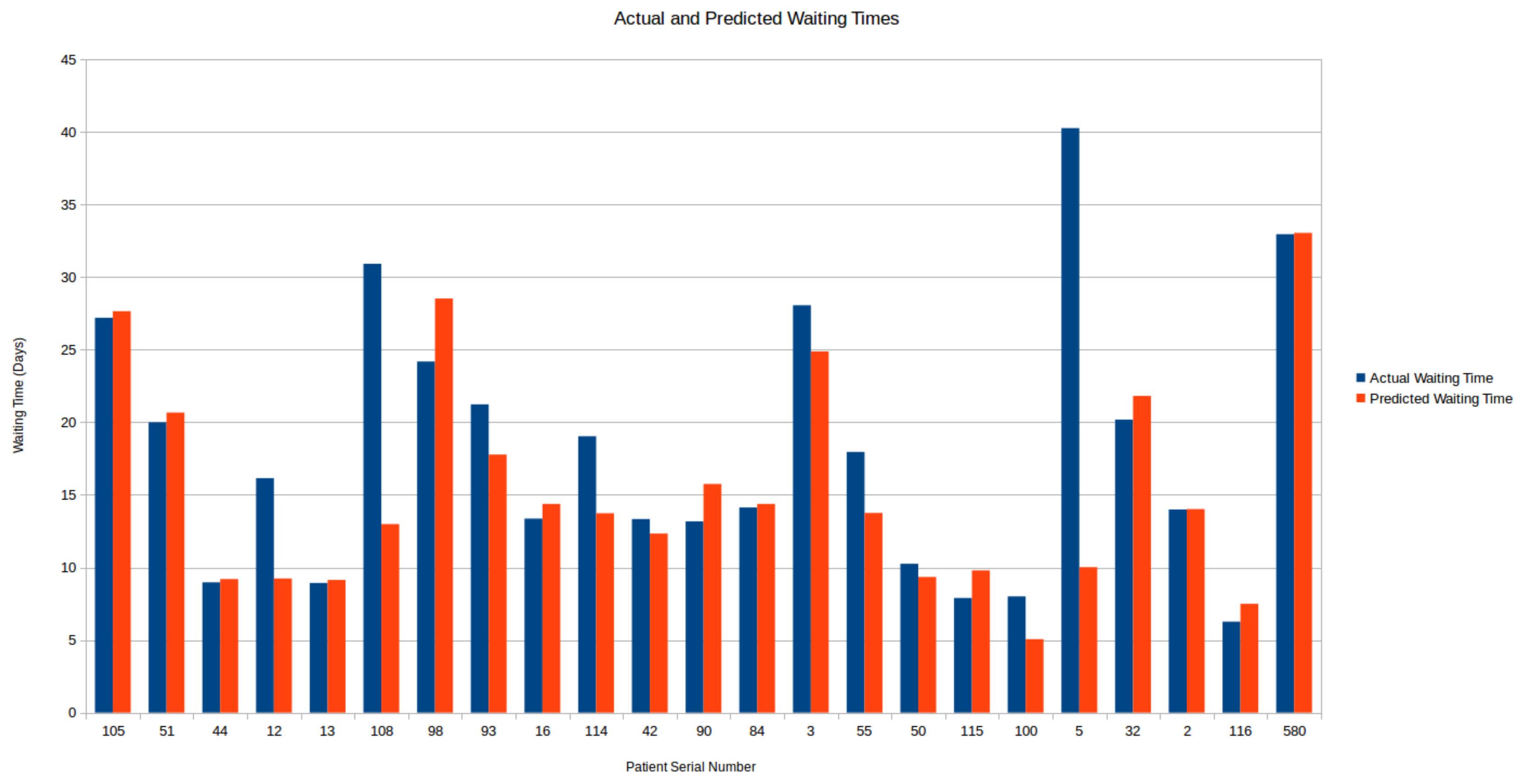


Figure 11: Comparison between the predicted waiting time with the algorithm and the actual waiting time from the data set.

8 Conclusions and Future Work

The contributions of this paper are as follows: 1) Establishing a more intuitive way of working with a data set through ORM. 2) Creating visualizations which are intended to provide invaluable information for both patients and doctors regarding the waiting times. 3) Creating a web application which combines the two previous points to have a universally accessible and data driven portal to visualize live data. 4) Approaching the problem of the waiting time to break the total wait into the discrete stages. 5) Using predictions with the staging to provide on-the-fly predictions given stages completed.

The results of the algorithm prediction on the 1/5 data set random patients leaves much to be desired. Although there was a 3.5 day average from the 21 patients in Figure 11, on the most part, the prediction doesn't work that much better than the standard "2 weeks". The greatest two contributions to these results involve the lack of data and the over-fitting of the models chosen for the machine learning. With a lack of patients in the final results, there is clearly a large standard deviation in the data set. For instance, let us take the paper mentioned earlier where breast cancer was being predicted with machine learning. The researchers had 6291 patients to use. The data set in this paper, conversely, uses 2602 patients with full data. With more data, perhaps from the past but with filtered information for missing or invalid points, this prediction could get much better. Otherwise, an accurate algorithm will not be able to perform proper predictions until more data is collected in the upcoming years of research.

Over fitting of data is a major concern because the problem becomes two-fold. At first, the algorithm seems to perform very well based on the training data set and cross-validation. However, with an expansive data set, it is clear to see that this is not the case. As long as there is data which is incomplete or deviates from the training data, the algorithm cannot preform well. This seems to be the case for all of the algorithms explored in this paper. In order to mitigate the effect of over-fitting, a more expansive data set can be obtained, with yet more patients, as well as looking at factors which are more broad and don't wrap the model too tightly around the data which is available.

In terms of deploying parts or all of this project for future use, there is great optimism. The prediction algorithm can be applied to see the remaining time left in the timeline view, as in Figure 11 but for individual patients. This would work well in the vision of having mobile applications for patients and doctors alike. Furthermore, visualizations like those presented in this paper can give clinicians a quick view of what is happening in the clinic without having to work through big blocks of text and other data. This can be implemented on screens through the clinic and have a lasting effect as soon as possible. With both directions, this is not the end of the story for the prediction of waiting times. As more data comes in, especially with the advancements at the new MUHC site at the Glen, more data, in terms of both quantity and quality, will be available to continue to make strides.

Bibliography

1. Leung, Alvin. "Analyzing Radiation Oncology Data for Prediction of Radiotherapy Patient Wait Time." (2014): n. pag. Web.

2. Sawaf, Marya. "How Much Longer? Predicting Waiting Times in Hospital Waiting Rooms." (2014): n. pag. Web.

3. Danjoux, C., et al. "An innovative rapid response radiotherapy program to reduce waiting time for palliative radiotherapy." *Supportive care in cancer 14.1* (2006): 38-43.

4. bin Othman, Mohd Fauzi, and Thomas Moh Shan Yau. "Comparison of different classification techniques using WEKA for breast cancer." *3rd Kuala Lumpur International Conference on Biomedical Engineering 2006.* Springer Berlin Heidelberg, 2007.

5. Wu, Qinglin, Yanzhong Hu, and Yan Wang. "Research on Data Persistence Layer Based on Hibernate Framework." *Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on.* IEEE, 2010.

6. Holmes, Geoffrey, Andrew Donkin, and Ian H. Witten. "Weka: A machine learning workbench." *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on.* IEEE, 1994.

7. Rasmussen, Carl Edward. "Gaussian processes for machine learning." (2006).

Appendices

1 Appendix: ORM Modelling

An example of the requirements for a table to be translated to an object in JAVA follows. Note that first, each column in the table must be set as a field in the object. These fields must have the same type as the corresponding column. Named queries can be specified at the start of the file and describe a way for data to be retrieved for a specific field. The second file, *persistence.xml*, contains the database connection information and specifies where the entities, or relation mapping files, lie in the source.

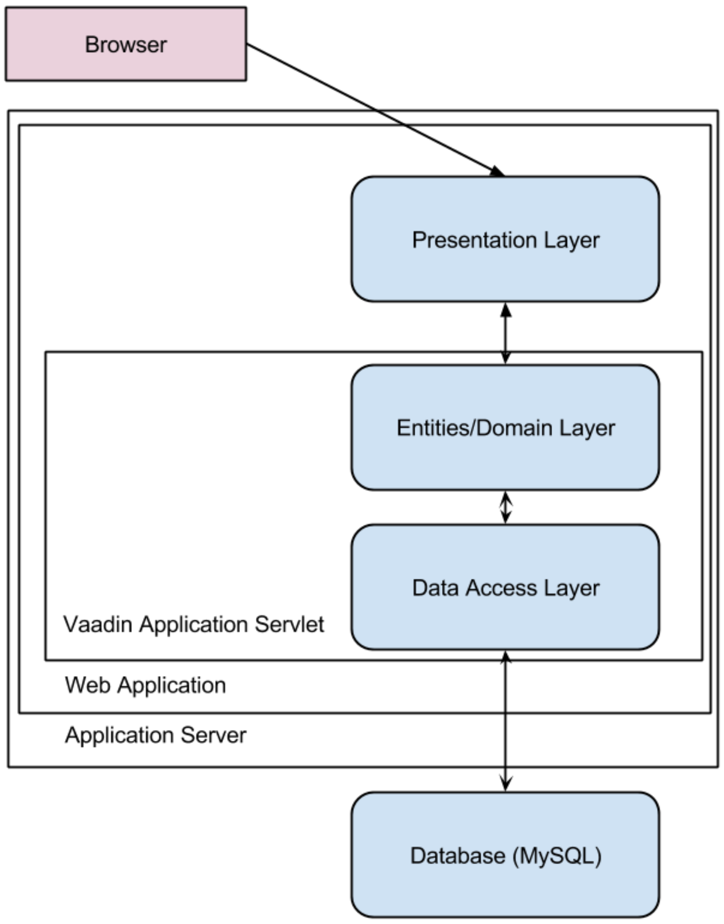
/src/main/java/com/mgorshkov/hig/business/entities/Alias.java

```
package com.mgorshkov.hig.business.entities;

import javax.persistence.*;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
import java.io.Serializable;
import java.util.Date;

@Entity
@Table(catalog = "hig20150218", schema = "")
@XmlRootElement
@NamedQueries({
        @NamedQuery(name = "Alias.findAll", query = "SELECT a FROM Alias a"),
})
public class Alias implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(nullable = false)
    private Integer AliasSerNum;
    @Size(max = 100)
    @Column(length = 100, nullable = false)
    private String AliasName;
    @Size(max = 25)
    @Column(length = 25, nullable = false)
    private String AliasType;
    @Size(max = 11)
    @Column(length = 11, nullable = false)
    private Integer AliasUpdate;
    @Column(nullable = false)
    private Date LastUpdated;

    public Alias(){}
    public Integer getAliasSerNum() {
        return AliasSerNum;
    }
    public void setAliasSerNum(Integer aliasSerNum) {
        AliasSerNum = aliasSerNum;
    }
    public String getAliasType() {
        return AliasType;
    }
    public void setAliasType(String aliasType) {
        AliasType = aliasType;
    }
    public String getAliasName() {
        return AliasName;
    }
    public void setAliasName(String aliasName) {
        AliasName = aliasName;
    }
    public Integer getAliasUpdate() {
        return AliasUpdate;
    }
    public void setAliasUpdate(Integer aliasUpdate) {
        AliasUpdate = aliasUpdate;
    }
    public Date getLastUpdated() {
        return LastUpdated;
    }
    public void setLastUpdated(Date lastUpdated) {
        LastUpdated = lastUpdated;
    }
}
```

/src/main/resources/META-INF/persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0"
             xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
             http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
    <persistence-unit name="hig20150218">
        <provider>org.hibernate.ejb.HibernatePersistence</provider>
        <jta-data-source>java:jboss/datasources/hig20150218</jta-data-source>
        <class>com.mgorshkov.hig.business.entities.Alias</class>
        <exclude-unlisted-classes>false</exclude-unlisted-classes>
        <properties>
            <property name="hibernate.hbm2ddl.auto" value="update" />
            <property name="hibernate.show_sql" value="false" />
            <property name="hibernate.max_fetch_depth" value="0" />
            <property name="hibernate.cache.use_second_level_cache" value="false"/>
            <property name="hibernate.cache.use_query_cache" value="false" />
        </properties>
    </persistence-unit>
</persistence>
```

2 Appendix: Software Structure



In brief the architecture of the web application is described above. In terms of technologies used: the application server is WildFly 8.2.10. The database used is MySQL. The JAVA framework which provides the presentation layer as well as the RCP communication between that and the data layer is Vaadin. The data access layer is mapped to the domain layer using Hibernate ORM. All data processing, and presentation is done in pure Java. All analysis for machine learning is done with the WEKA toolkit.