Bayer Pressure Predictor Project

Foundations of BME Design 42-401
Carnegie Mellon University

Submitted by:
Anusha Agarwala
Gage Anderson
Shayan Gupta
Mia Han
Sean Pereira

Advising TA:
Melanie Loppnow

Sponsor: Bayer Corporation
Sponsor Contact: John Volkar

1. Abstract

More than 70 million Computed Tomography (CT) scans are conducted in America per year [15]. In each CT scan, the CT contrast agent, typically iodine, is injected into the patient. Simultaneously, the x-ray imaging takes place. To achieve ideal resolution for CT scans, a specific injection protocol is set to control the flow rates and volumes of the fluids delivered to the patients. CT injections are typically delivered at flow rates of 2-7 mL/s and have viscosities between 10 and 30 cp that vary with temperature. The catheter type and size has an impact on the way that fluid is delivered to the patient as well. The fluid within the CT injector is typically two phases, consisting of contrast and saline. The catheters have a maximum pressure rating that cannot be exceeded. Exceeding this maximum pressure rating leads to a pressure limiting scenario (PLS), where the flow rate is reduced to bring the catheter back under the pressure limit. This is nonideal, because a lower flow rate of contrast agent delivered at a point in time when the x-ray is imaging can result in a lower resolution CT scan. Under enhanced images are non-diagnostic, wasting treatment time and resources for the radiologist, patient, and physician. Another risk possible from exceeding the pressure rating is a pressure disarm scenario (PDS) in which the system aborts to counteract the exceeding of the maximum pressure. This project focuses on creating pressure prediction curves using past injection data and curves. From this information, injection and scan protocols can be programmed to ensure proper delivery of contrast without exceeding the pressure limit. The final product is a software tool that includes visual or audio alerts when injector configurations will result in a PLS or PDS.

2. Description of the Problem

When patients are prepared for a CT scan, there is currently no standardization of protocols for the CT Contrast Injector that the technicians follow. Technicians are trained by Bayer specialists, but the configuration of the CT Injector is completely up to the technician based on their prior experiences or educated guesses, and based on the patient's characteristics like height, gender, age, and weight. The injector settings vary from technician to technician and there is no standard method of configuring the pressure settings. Moreover, technicians currently are unable to determine whether the injection procedure parameters (protocol) they configured will reach a pressure-limiting scenario (PLS) or pressure-disarming scenario (PDS). Thus, our goal is to determine if a protocol will result in a PLS or PDS. With the proposed CT Injection Pressure Predictor, technicians will be able to use our predictor to determine if a configuration is incorrectly set up for that patient prior to actually injecting them with contrast. This software tool will therefore help CT technologists and radiologists maximize the effectiveness of CT scans so that they have fewer failed injections and minimize the ration exposure to the patient and themselves.

There are two main purposes, or "must-haves," that are essential to our proposed solution. One purpose is that our solution has to inform technicians if their selected procedure

parameters (such as the catheter type, max flow rate) will result in a PLS or PDS before the start of the injection. Another necessary feature is that our solution must be able to interface with the Stellant® CT injector display system because the technicians already refer to this display before and during the procedure.

There are also a few features, or "nice-to-haves," that are helpful but not indispensable. One helpful feature is giving the technician suggestions (i.e. changing the catheter to one with a higher pressure rating) on attaining the desired pressure scenario. Another helpful function is to alert the technician of undesired pressure scenarios *during* the procedure. This function would be useful in getting the technician's attention, but often there is little the technician can do to remedy the situation at that point. A third helpful function is to interface with the Stellant® pressure sensors to collect real-time pressure data. This data could be used to improve future predictions and help give real-time alerts.

3. Needs Objective Statement

"A software tool that will analyze contrast injector settings and use previous injector data to predict whether a particular configuration will result in failure of contrast injection during a CT scan." This will allow radiologists to ensure that they can get a successful scan before wasting time and resources.

4. Documentation of the Design

As mentioned in our goals for the software tool, the software must be able to predict, based on data from previous injection procedures, the maximum pressure values for a procedure given the technician's inputted protocol. We have taken two approaches: a mathematical model and a machine learning model. The mathematical model requires us to understand the underlying physics of flow through the catheter during a contrast procedure and is more precise. The machine learning model has an increased accuracy, but is limited in precision and it is difficult to ascertain the underlying model computations.

a. Mathematical Model

The mathematical model's design is structured into three segments: a data parsing program, a nonlinear analysis program, and an output program. The data parsing program takes in the formatted text file provided to us by Bayer and outputs the time, pressure, and flow rate data for each injection sorted into buckets by the catheter gauge, contrast type, and saline type, as shown in Figure 1. The nonlinear analysis program takes in all the pressure vs. time and flow rate vs. time graphs from a given bucket (same gauge, contrast type, and saline type) and outputs an empirical model. This model

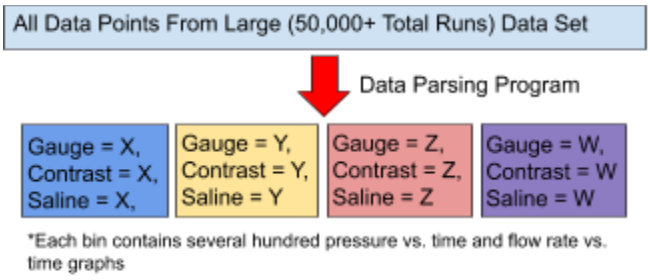

Figure 1. Parsing existing injection data into buckets.

is generated using Poiseuille's Equation for laminar flow scaled by nonlinear factors that are defined to best fit the data from previous trials. The final program graphs the data and would send the empirical model to the injector's display system for the technician to see the predicted output curve. The output of the program is a plot similar to the MATLAB plot shown in Figure 2. The finished product would be able to export this graph to the Stellant system in order to display the result to a technician.

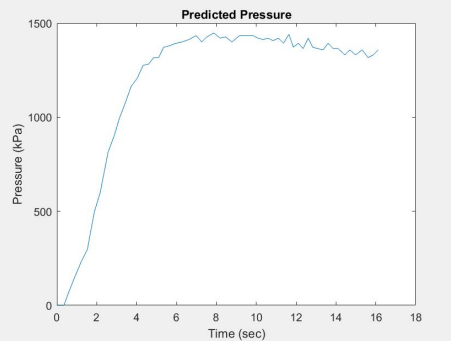

Figure 2. Sample output pressure graph.

b. Machine Learning Model- Neural Network

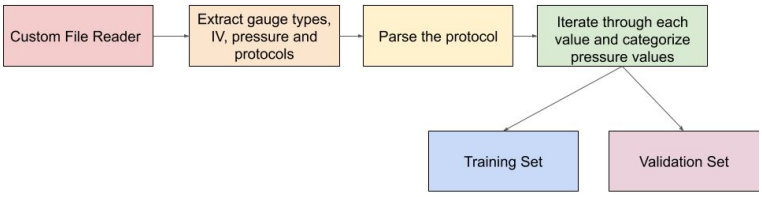

Figure 3. Data Processing Pipeline

The first step of the machine learning model was parsing the data and sorting it into testing and training sets. The definition of these sets are done in the data processing pipeline described below and depicted in Figure 3.

Data Processing Pipeline

1. **Custom file reader:** The raw data is first processed by a CSV file reader that removes the extraneous characters and whitespace from the file. The output is the original data with irrelevant characters and markings removed.
2. **Extract relevant features:** Irrelevant data (such as trials where the user "aborts" before completion) is removed and cleaned from the data set; contrast, saline, catheter type, IV, gauges, location, placement,/ protocol, pressure and whether or not the injection was terminated are the variables that we used. The output is an entire CSV file converted into a 2-D array, where each item in the array represents an entry in the table
3. **Parse the protocol:** The protocols are then further parsed and enumerated; the values are converted into appropriate types and formats for the neural network to use.
4. **Iterate through each value and categorize pressure values:** The data is then placed into "buckets"; a "bucket" is a value range that the data falls into (i.e buckets for pressure values are a bucket for the 0-20 kpa pressure values, a bucket for 20-40 kpa pressure values, etc). The "bucket" is a list for a specific range of pressure limit values. The output is a fully processed and parsed data set ready to be inputted into the neural network.
5. **Randomly dividing up the data set:** The dataset is then split into two halves: a training set used to train the model and a validation set used to verify the model's accuracy. Each entry in the dataset is randomly selected into either the training and validation set to ensure that the network won't be affected by underlying bias.

Once training sets and testing sets were established, training of the neural network could begin. The network architecture is depicted in Figure 4 and described below.

Neural Network

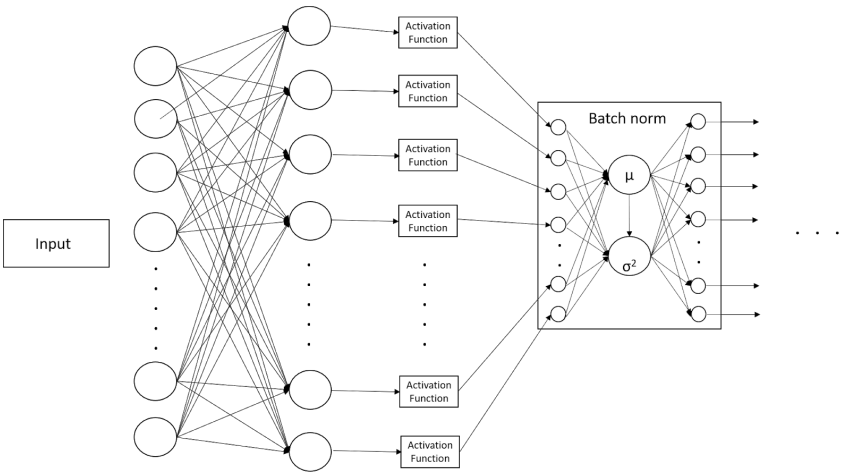

Figure 4. Depiction of a Neural Network.

The cleaned data (input) is categorized into a bucket[1] (circles); the different columns of circles in the image represent the layers of the neural network. The layers of a neural network consist of perceptrons[2]. For the model, the optimal number of layers is three layers; after testing various values, using three layers yielded the highest accurate prediction rate.

The first layer of the neural network is then passed through an activation function[3]; the neural network uses rectified linear unit activation functions. The output of the activation function is then batch normalized[4]. The process is repeated for the second layer of the neural network. Once the neural network is trained, the second set of data, the validation set, validates and measures the accuracy rate of the neural network.

5. Prototype of the final Design

The prototype of our final design is a GUI that encapsulates and compares the two approaches to predicting the pressure curve and whether or not a given configuration will yield a successful injection: through the GUI , users can simulate both approaches without needing to manually open the various files and run them from their terminal. The GUI will show users the graphs that illustrate both models as well as the percentage of accurate classification. Our two approaches consist of:

1. Neural Network: The neural network approach consists of two different models: the regression based neural network and recurrent based neural network.

[1] A "bucket" is a value range that the data falls into (i.e buckets for pressure values are a bucket for the 0-20 kpa pressure values, a bucket for 20-40 kpa pressure values, etc)
[2] Perceptron is a neural network unit that does certain types of computations to extract features from the data set
[3] An activation function is the first layer of the neural network and a type of transfer function that determines the output of the neural network
[4] Batch normalization is a technique used to minimize the data distribution shifts as the parameters of the layer are changing

2. Mathematical Model: Uses Poiseuille's model and step-wise functions to generate a pressure prediction curve.

Description of the GUI

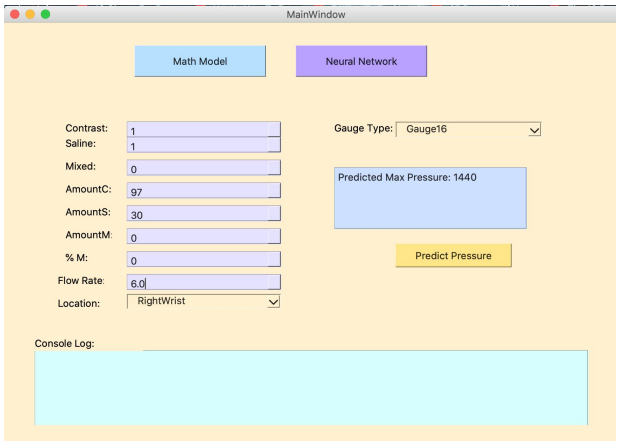

Figure (Left): A screenshot of the GUI on the neural network page. Given an inputted protocol configuration (consisting of ten different input fields), the GUI will determine the maximum pressure.

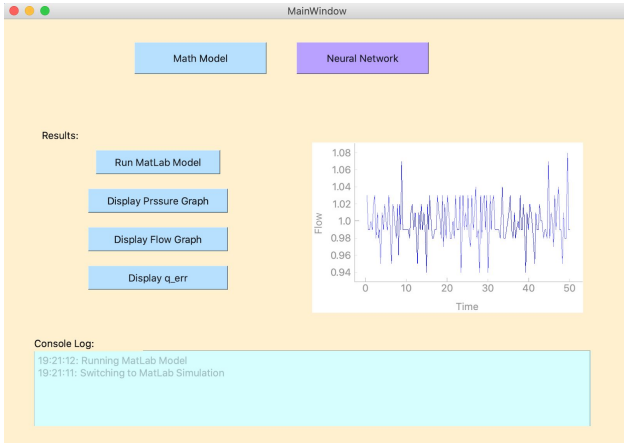

Figure (Right): A screenshot of the GUI on the MatLab page. Because the MatLab model generates a prediction of the line graph, it does not take any data in as inputs. Therefore, the user can click the simulate button to output the different predicted curves.

The GUI is a simple interface implemented in Python that wraps around the models to display the outputs and the functionality in an organized way. The GUI consists of two screens: the MatLab simulation and the Neural Network model simulation. The user can toggle between both screens by clicking the top center buttons to compare the results; the console log updates every time the user clicks a button on the screen. After the user enters a protocol configuration on the neural network page, the GUI will output the maximum pressure determined by the neural network; to simulate another protocol, the user can enter another protocol configuration and click the "Predict Pressure" button. Because the MatLab model does not take in any inputs, the user can simply click the run MatLab model simulation to visualize the predicted pressure curve.

Overall Design

Neural Network Architecture

Using a neural network with three layers, as depicted in Figure 4, yielded the highest accuracy rate for pressure prediction. Likewise, after trying various layer types like bias layer, linear layer, and linear-feed

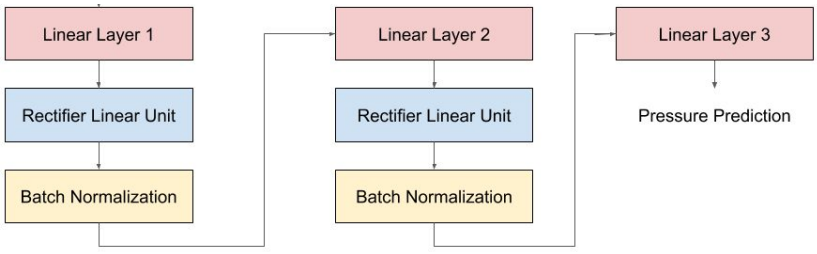

Figure 4. Neural Network with 3 convolution layers.

forward layer, using a linear layer[5] yielded the best results. Batch normalization[6] ensured consistent outputs, stabilized the training, and reduced the latency to output results. Moreover, a rectifier layer unit[7] is a standard activation function most often used with multi-layer neural networks. More details about the neural network layers are located in the Appendix.

Data Specifications

1. **Input:** Features of a given protocol provided by the technician
2. **BatchSize[8]:** The BatchSize used was 256.
3. **Output:** Probability that a given protocol was in a certain pressure "bucket" (+/- 20 kpa).

Future Designs/Currently Working On

1. **Regression Based Model:** A similar model to the previous model, the regression based neural network model consists of a different output linear layer and uses a different type of loss function. Unlike the previous model's output (multiple buckets), the neural network outputs a single output used to extract the maximum pressure. The model uses a different loss function, Mean Squared Error instead of Cross Entropy Loss; Mean squared error (MSE) is the most commonly used loss function for regression. The loss is the mean overseen data of the squared differences between tru.
2. **Recurrent Based Model:** A recurrent based model is structured completely differently from the previous two models; it takes in an input and a hidden state, where input is at a given time step. The hidden state is information that has been "saved" from prior time steps. The model will instead output a pressure prediction graph, while the previous two models outputted a single pressure prediction for each entry.

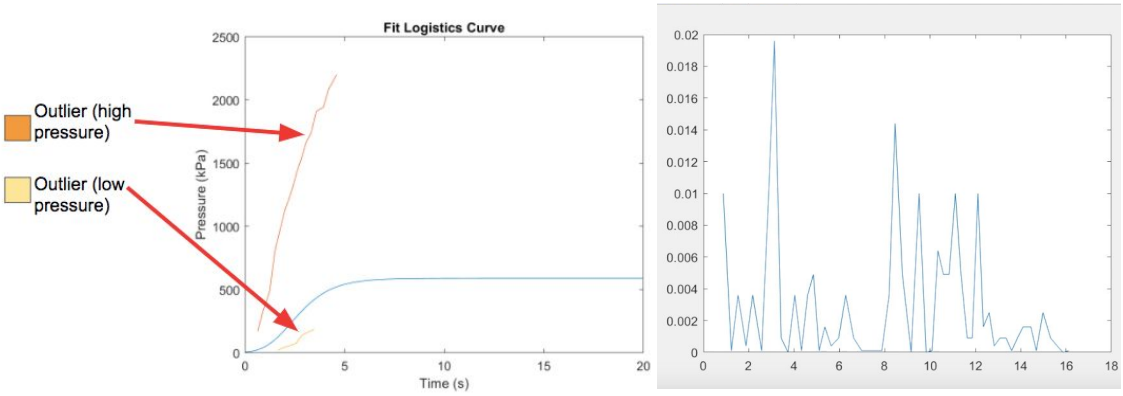

Figure 5: Logistics curve (left) does not fit pressure data, and error plot from step model shows all error less than 2%.

MatLab Architecture

After testing a variety of different statistical models with over 5% error, a step model with a low-pass filter proved to be the most effective at providing <2%

[5] A linear layer is a layer without bias and capable of learning an average rate of correlation
[6] Batch normalization is a technique used for training neural networks that standardizes the inputs to the layers and provides regulation of outputs
[7] Rectified Linear Unit is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero and most commonly used as activation function.
[8] Batch size is the number of training samples processed before the model is updated

error for approximately 80% of the data. The large data file was parsed so that data with "user abort" was removed, and the remainder of the data was sorted into vectors based on gauge type. The logistics model was not a valid model of the data, whereas the step model was a valid model, as shown by figure 5.

6. Proof that the design is functional and will solve the problem

a. Machine Learning Model- Neural Network

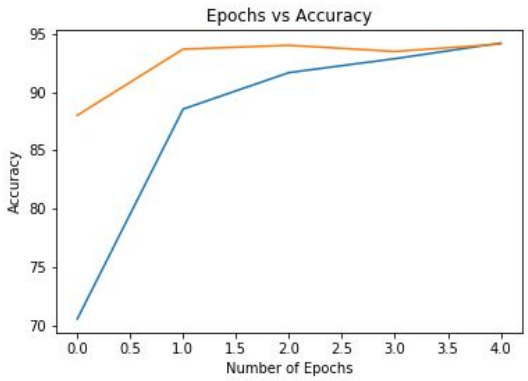

Figure 6: The blue line represents the accuracy achieved on the training data, and the orange line represents the accuracy achieved by the validation data.
An epoch is one cycle through the training set, and the accuracy is the percentage of maximum pressure values that is predicted correctly. As the neural network increases the number of times that it trains on the model, it learns more about the data, ultimately yielding a 95% accuracy.

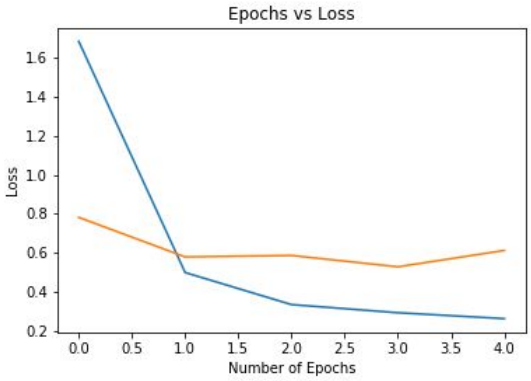

Figure 7: The blue line represents the accuracy achieved on the training data, and the orange line represents the accuracy achieved by the validation data.
Loss is the result of a bad prediction and indicates how bad the prediction; ideally, the loss value should be close to zero. As the neural network increases the number of times that it trains on the model, it is better trained and able to make better predictions, causing a decrease in the loss value

For each epoch the training dataset and validation dataset was shuffled (randomized); the overall dataset was split 40%/60% with training and validation respectively. The overall goal of a machine learning model is to learn from the training dataset by generalizing the features from the training dataset. Thus, the validation dataset provides us with feedback on the model's ability to make generalizations. The loss and accuracy of the validation dataset should be proportional with the training dataset. If it doesn't follow this trend, the model will start to overfit. Regularization techniques are usually used to prevent this overfitting such as batchnorm. As shown in Figure 6, the validation set data was able to achieve the same ~95% accuracy as the training set (left graph) while not overfitting the data which would result in loss as low as that of the training set (right graph).

b. Mathematical model

One approach to solving this problem mathematically is to preprocess the data so that all trailing zeros and leading zeros are removed. This way only the contrast phase is considered. After cutting down the data to this phase, the data can be fit according to Poiseuille's formula (equation 1). For the Poiseuille variables that need to be optimized, since they are unknown or

could vary between procedures, we have been able to lump them into one variable because they're all linear according to Poiseuille. By visual inspection, a first-order low pass filter was added because the pressure rises slower than the flow rate does. The variable that parametrizes the low pass cutoff was optimized. In this approach, the model is dependent on the flow-rate data presented by Bayer.

In order to create a model that would predict the flow rate, and therefore the pressure, a stepwise function will be utilized. When tested on all gauge types, this statistical model produced an error of less than 5%, disregarding noise in data.

Equation 1: $\Delta p = \frac{8\mu L Q}{\pi R^4}$ where L is the length of the pipe, Q is the volumetric flow rate, μ is the dynamic viscosity of the liquid, and R is the pipe radius.

7. Results of Patent Search

There exists no patents for deep learning models that characterizes protocols to pressure Ratings. However, patents do exist for Matlab and Neural Networks albeit no existing prior art for the purpose of modeling contrast injection. By producing a GUI as a final prototype that illustrates graphs for both models, we are producing a novel approach to determining the protocol for injectors. The methodology of our prototype is patentable, unlike purely computational software, because the prototype takes user input and presents the projection as a part of the GUI. Thus the requirements for this to be a patent do exist. Our prototype is useful and non-obvious, otherwise there would not have been a need for such software. This evaluation is useful for our client as they can leverage this unique IP to boost sales of their injectors.

8. Anticipated Regulatory Pathway

Our software is not intended to diagnose, cure, mitigate, treat, or prevent disease, so the software is not classified as a medical device by definition. More specifically, the software is not directly changing the injection procedure. It is simply informing the technologist's parameterization. Because the software provides assistance to the technician or health care provider (HCP), it falls under *Non-Device Clinical Decision Support (CDS)*. According to the Clinical Decision Support Draft Guidance, Non-Device CDS must meet four requirements: (1) not intended to acquire, process, or analyze a medical image or a signal from an *in vitro* diagnostic device or a pattern or signal from a signal acquisition system, (2) intended for the purpose of displaying, analyzing, or printing medical information about a patient or other medical information, (3) intended for the purpose of supporting or providing recommendations to an HCP about prevention, diagnosis, or treatment of a disease or condition, and (4) intended for the purpose of enabling an HCP to independently review the basis for the recommendations that such software presents so that it is not the intent that the HCP rely primarily on any of such recommendations to make a clinical diagnosis or treatment decision regarding an individual patient [17].

Our software fulfills the first requirement because it does not process any signals from an *in vitro* diagnostic device or any otherwise acquired signal. The software fulfills the second requirement because it is intended to display the possible pressure graph of the simulated injection. The software fulfills the third requirement because it is intended to support the technician, the HCP's, decisions regarding the contrast injection procedure, which is used for diagnosis of a health condition. The software fulfills the fourth requirement because the HCP does not have to rely on the software because any recommendations the software provides as per the risk of reaching a high-pressure scenario can be independently reviewed by the HCP. In other words, the HCP can fulfill the contrast injection procedure if he/she disregards any recommendations from our software.

Our Non-Device CDS will be under enforcement discretion, meaning that FDA will not require any premarket notification. It is highly advisable to have post-market monitoring to ensure that there are no/limited safety and effectiveness issues. FDA will also use this marketing to justify enforcement if the software is not used as a purely informational tool. To supplement the quality and robustness of our software, we will follow the recommendations set forth in the *ISO/TS 13972:2015* standard concerning detailed clinical models, characteristics and processes. This standard defines Detailed Clinical Models (DCMs) in terms of an underlying logical model of clinical concepts that can be used to define and to structure clinical information, such as contrast injection procedure parameters [18]. We will also follow *IEEE 1012-2016*, which prescribes the necessary actions to verify and validate software. This verification and validation is key to ensuring that our software tool works uniformly across all Stellant injectors with all possible technician input configurations [19].

9. Reimbursement

Our software is not a standalone technology. Its purpose is to inform the technician's contrast injection procedure. Our software does not directly change the procedure in any way, so existing payment codes for the contrast injection procedure will be used by the technician for the procedure. Existing payment codes for the contrast itself and procedure will continue to be used. The codes comprise: *A9698* (Non-radioactive contrast imaging material, not otherwise classified, per study) and codes *Q9958-65* (Osmolar contrast material, based on iodine concentration in mg/ml, charged per ml). As a result, Bayer will not need to receive a new payment code from the Center for Medicare and Medicaid Services. By extension, there will be no change in how insurance providers cover the medical costs associated with the contrast injection procedure.

10. Estimated Manufacturing Costs

The estimated manufacturing cost of adding this software to CT contrast injectors will vary. The primary cost of software production comes from its development. In other words, these costs arise in a one-time fashion, including the cost of developing the software to integrate into the existing Stellant system as well as testing for software verification and validation as well as

human factors. If the Stellant device to which the program will be uploaded already has space for this software, the amortized cost of adding this software to every such Stellant injector is near $0. But, if the device doesn't have a sufficient amount of memory or the proper hardware requirements to run the network, the manufacturer may need to add more memory to the device or upgrade the cpu. The cost depends completely on the amount of memory and the type of cmu the manufacturer would like to add. For minimal upgrade requirements, the estimated cost would be between $80.00 to $160.00. Because the devices are constantly being modified and vary, the costs using our algorithm vary depending between different injector system versions and manufacturers. The manufacturing cost would be the same as only the parts would change.

The hardware of each injector needs no updating since this software is able to run by simply recording the specifications of the procedure and catheter used in the injection. It is important to note that the model would have to be re-distributed every few months with updated procedure data. This process, however, has no direct manufacturing cost because re-training the model on a remote server is a one-time cost.

11. Potential Market and Impact

As of 2018, the global contrast media market was expected to grow about 4% annually with an expected market valuation of at least $5.65 Billion by 2022 [11]. The drivers of this growth deal with the patients and the technology. With an increase and growth in contrast media usage, the demand for medical imaging and scans will continue to surge, leading to an increase of contrast injectors that could utilize our prediction models.

One of the customers of our prediction models would be the hospitals that purchase contrast injector systems from Bayer, specifically Bayer's Stellant injector[9] and Ultravist iodine contrast [10]. The injector and contrast are used for radiology and cardiology applications, are iodinated products, are generally administered intravenously, are used in CT and MRI procedures, and have cardiovascular indications. Contrast injector systems enable physicians to configure the correct injection settings, but oftentimes, the injection may fail and not successfully inject due to the misconfiguration. With our prediction models integrated with their current injector system, technicians would be able to more accurately inject contrast into the patient, and thus yield a higher injection success rate. Since the prototype is currently yielding at least a 95% accuracy rate, the model would allow technicians to more successfully inject CT contrast into patients, causing an increase in efficiency and reduction of wasted contrast.

[9] The Stellant D CT Injection System is a full-featured, dual syringe injection system that reportedly enables clinicians to perform the most critical CT contrast exams with advances like DualFlow simultaneous saline and contrast delivery

Appendix

Neural Network Architecture Details: **Features of the Neural Network:** In addition to the overall neural network structure previously discussed, neural networks may also contain optional optimization and loss functions to yield more accurate and faster results.

1. **Criterion:** Cross-entropy loss (log loss), measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label.
2. **Optimizer:** Adam Optimizer[10] is based on adaptive estimates of lower-order moments. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters. This is utilized to minimize our loss given our weights and biases.
3. **Scheduler:** Lastly, the model utlizes ReduceLROnPlateau[11] and benefits from reducing the learning rate by a factor of 2-10 once neural network learning stagnates. The scheduler reads a metrics quantity and if no improvement is seen for a 'patience' number of epochs, the learning rate is reduced.

By utilizing these features in the neural network prototype, the neural network yielded a higher accuracy rate and outputted results quicker than the initial implementation of bare-boned neural network.

Link to Final Video:
https://drive.google.com/file/d/1H3OTnHrsOldw40GFQvXWphIUzu4Oezby/view?usp=sharing

[10] Adam Optimizer is an algorithm for first-order gradient-based optimization of stochastic objective function
[11] Reduces learning rate when a metric has stopped improving

Bibliography

1. "What Is a Stroke?". www.nhlbi.nih.gov/. March 26, 2014. Archived from the original on 18 February 2015. Retrieved 26 February 2015.

2. "Who Is at Risk for a Stroke?". www.nhlbi.nih.gov. March 26, 2014. Archived from the original on 27 February 2015. Retrieved 27 February 2015.

3. "How Is a Stroke Diagnosed?". www.nhlbi.nih.gov. March 26, 2014. Archived from the original on 27 February 2015. Retrieved 27 February 2015

4. Adams HP, Bendixen BH, Kappelle LJ, Biller J, Love BB, Gordon DL, Marsh EE (January 1993). "Classification of subtype of acute ischemic stroke. Definitions for use in a multicenter clinical trial. TOAST. Trial of Org 10172 in Acute Stroke Treatment". Stroke. 24 (1): 35–41.

5. Anonymous (2014-07-29). "Hemorrhagic stroke". National Stroke Association. Archived from the original on 27 June 2016. Retrieved 30 June 2016.

6. Hill MD (November 2005). "Diagnostic biomarkers for stroke: a stroke neurologist's perspective". Clinical Chemistry. 51 (11): 2001–2.

7. Bamford J, Sandercock P, Dennis M, Burn J, Warlow C (June 1991). "Classification and natural history of clinically identifiable subtypes of cerebral infarction". Lancet. 337 (8756): 1521–6.

8. Feigin VL, Rinkel GJ, Lawes CM, Algra A, Bennett DA, van Gijn J, Anderson CS (December 2005). "Risk factors for subarachnoid hemorrhage: an updated systematic review of epidemiological studies". Stroke. 36 (12): 2773–80.

9. Antonio Di Carlo, Human and economic burden of stroke, Age and Ageing, Volume 38, Issue 1, January 2009, Pages 4–5, https://doi.org/10.1093/ageing/afn282

10. “Bayer in Radiology Products.” Bayer Radiology, www.radiologysolutions.bayer.com/products.

11. “Contrast Agents Market Is Expected to Exceed US$ 5.65 Billion By 2022.” *MarketWatch*, 17 Aug. 2018, ww.marketwatch.com/press-release/contrast-agents- market-is-expected-to-exceed-us-565-billion-by-2022-2018-08-17.

12. “MEDRAD® Stellant CT Injection System: Radiology US.” Bayer in Radiology Official Site, www.radiologysolutions.bayer.com/products/injection-systems/medrad-stellant-ct.

13. A Guide to an Efficient Way to Build Neural Network ... https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-i-hyper-parameter-8129009f131b.

14. Xing, Wanli, and Dongping Du. “Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention.” Journal of Educational Computing Research, vol. 57, no. 3, 2018, pp. 547–570., doi:10.1177/0735633118757015.

15. Harvard Health Publishing. “Do CT Scans Cause Cancer?” Harvard Health, Mar. 2013, www.health.harvard.edu/staying-healthy/do-ct-scans-cause-cancer.

16. “Symptoms of a Stroke.” Www.goredforwomen.org, www.goredforwomen.org/en/about-heart-disease-in-women/signs-and-symptoms-in-women/symptoms-of-a-stroke.

17. United States, Food and Drug Administration, Center for Device and Regulatory Health. “Clinical Decision Support Software Draft Guidance for Industry and Food and Drug Administration Staff.” *Clinical Decision Support Software Draft Guidance for Industry and Food and Drug Administration Staff*.

18. “ISO/TS 13972:2015.” iso.org, International Organization for Standardization, Oct. 2015, www.iso.org/standard/62416.html.

19. “IEEE Standard for System, Software, and Hardware Verification and Validation," in IEEE Std 1012-2016 (Revision of IEEE Std 1012-2012/ Incorporates IEEE Std 1012-2016/Cor1-2017), pp.1-260, 29 Sept. 2017