Anatomically personalized ML model predicts target temperature in focused ultrasound brain treatments

By: Chloe Ulsh, Charlotte Edwards, Jayati Maram

Word Count: 4572

Number of Figures and Tables: 8

Number of Equations: 0

Number of Supplements: 2

Number of Citations: 20

Approved_____Rick Hamilton_____Date_05/06/2024

Approved: Rithika Kormath Anand

Date: 05/06/2024

Anatomically personalized ML model predicts target temperature in focused ultrasound brain treatments

Chloe Ulsh[a], Charlotte Edwards[a,1], Jayati Maram[a], Samarth Singh[b], Rithika Anand[b], Rick Hamilton[b]

[a] Department of Biomedical Engineering, University of Virginia
[b] Focused Ultrasound Foundation
[1] Correspondence: cre3nue@virginia.edu

Abstract

Focused ultrasound can treat a variety of diseases through targeted, nonsurgical tissue destruction. For brain diseases, focused ultrasound treatment aims to thermally ablate an area of the brain by raising target tissue temperature above 55℃. Successful ablation is not guaranteed, as individual patient skull parameters and transducer settings affect the ability of the ultrasound beams to ablate. To predict ablation success before treatment, a skull density ratio is currently used, despite historical inaccuracies in predicting treatment success. The ability to directly predict the temperature a target can reach prior to a focused ultrasound treatment is vital to save time in the procedure, ensuring eligible patients benefit and reducing the discomfort of ineligible patients. Current methods do not accurately or simply predict the target temperature that will be reached during treatment. To predict the target temperature, a machine-learning model was built and trained using patient data. The model is a boosted tree regressor, called XGBoost using individual patient and transducer parameters. The final model has an R^2 value of 0.89 and an RMSE of 1.34℃. Part of the performance evaluation includes a decision tree for visualization and an analysis of the variance, both vital for future work on this model. To increase accessibility, a GUI is built using the model so researchers and physicians can input values and receive a prediction. The model can be improved in the future using more data and higher-resolution biological data, such as vasculature imaging of the brain. The final model's ability to make accurate pre-treatment predictions accelerates the adoption of, allowing more patients to to benefit from, this incisionless treatment option for brain diseases.

Keywords: high-intensity focused ultrasound, temperature prediction, machine learning, gradient-boosted random forest, precision medicine

Introduction

Focused Ultrasound Technology

Focused Ultrasound (FUS) is a noninvasive treatment method that allows targeted tissue destruction, or ablation. FUS tissue ablation in the brain is achieved by the combined heating effect of hundreds of individual high-intensity ultrasound beams generated from a helmet-shaped transducer and focused toward a common target. Applications of FUS to brain diseases are wide-ranging, spanning from neurological diseases such as Alzheimer's and Parkinson's to physical brain diseases such as brain tumors [1]. Besides pharmacological treatments, which are hindered by the blood-brain barrier, these disorders are often treated with radiotherapy, electroconvulsive therapy, or deep-brain stimulation. Compared to these alternatives, FUS is incisionless and extracorporeal, conferring rapid recovery time with minimal complications.

The US Food and Drug Administration first approved FUS for Essential Tremor in 2016, and more recently for advanced Parkinson's [2]. It is now actively being investigated for other disorders such as obsessive-compulsive disorder, chronic pain, epilepsy, and dystonia [3]. In contrast to the previously used surgical methods, the outcomes of FUS ablation are immediate and permanent. Therefore, it is imperative to select patients well, measure the precise location of targets, and carefully monitor patients throughout and after the treatments.

Current Pre-FUS-Treatment Metrics

Prior to a FUS treatment, computed tomography (CT) and magnetic resonance imaging (MRI) of the patient's head are collected and used for software-based simulations and calculations. The CT image is used to assess variation in the skull, the most significant attenuator of the ultrasound beams [4]. The MRI image is used to resolve the brain tissue structure [4].

After image collection, visualization softwares are used to plan the treatment and extract image-based calculations. One FUS visualization software is Kranion . This software is an open-source environment that allows users to simulate the FUS treatment as well as replay previous treatments [5,6]. When a treatment is loaded into the Kranion environment, the ray paths of the ultrasound can be viewed, illustrating how the transducer and the skull orientations impact the number of penetrating rays. Simulation and anatomical metrics can be exported from Kranion such as: the incident angle of the beam, the path length of refracted beams in the skull, skull thickness, and the skull density ratio. This list is not exhaustive but represents the skull metrics relevant to this project.

The skull density ratio (SDR) can be calculated in Kranion, which is especially useful for physicians since it is the primary metric with which physicians currently predict whether or not a focused ultrasound treatment will be successful for the patient [6]. The SDR represents the global averaged ratio between the mean Hounsfield unit values, a measurement of radio density usually used in the interpretation of CT images, between the skull's cancellous and cortical bones [7]. Values greater than 0.4 are considered more suitable for treatment [8]. Values less than 0.4 typically reflect a denser skull, meaning that due to greater attenuation and reflection of the ultrasound beams, more energy would be required to reach a therapeutic threshold temperature of 55℃ [4]. The SDR is limited by a high margin of error, where patient SDR scores do not historically accurately predict patient treatment success, excluding patients who could benefit from treatment and including patients who cannot be fully treated [7].

Temperature in Focused Ultrasound

The goal of a FUS treatment is to achieve thermal ablation of target tissue. Thermal ablation is heating sufficient to cause protein denaturation, and thus tissue death, in the target region. In a FUS treatment, because the combined heating effect of many ultrasound beams at the focus is required to cause ablation, tissues surrounding the target region are not ablated.

Thermal ablation is achievable through FUS because ultrasound beams are mechanical compressional waves. When a high-intensity ultrasound beam enters tissue, the tissue compresses and stretches with the beam. The friction of the relative movement of tissues converts the mechanical movement energy to heat energy. The combined heat energy generated at the target point by many high-intensity beams is sufficient to raise tissue temperature to the therapeutic threshold for FUS ablation, widely considered 55℃ [9]. The true temperature required to cause FUS ablation varies with exposure and is more accurately described using a cumulative exposure metric of cumulative equivalent minutes at 55℃, CEM55. CEM55 is modified from the CEM43 relationship for ablative radiofrequency exposure, which accounts for both time at a tissue temperature and the difference between the current temperature and the threshold [9].

Thermal ablation is distinct from hyperthermia, the result of achieving a tissue target temperature lower than 55℃. Hyperthermia increases cell permeability, allowing for greater blood flow and drug absorption without permanent damage [4]. Hyperthermia has been studied extensively and can increase the efficacy of chemotherapy and radiation therapy in solid tumors. FUS systems have both thermal ablation and hyperthermia properties. However, for localized neurological disorders like essential tremor, large lesions are necessary for treatment, which can only be created through thermal ablation.

After a target region is ablated, destroyed tissue is ultimately naturally removed by the body, making the FUS procedure wholly incisionless. The ability to remove diseased brain tissue without surgery, especially for target regions deep in the brain, makes research into methods for best achieving ablation temperatures through FUS extremely attractive.

Current Temperature Prediction Methods

Current FUS treatments require significant on-the-fly adjustments during treatment before successfully achieving thermal ablation temperatures. This is due to the fact that while pre-treatment planning data is collected, and methods exist to predict the temperature reached during a FUS treatment, these temperature prediction methods have not been sufficiently developed or clinically utilized to accurately predict a patient's treatment temperatures. Some temperature prediction methods use statistical methods to categorize patients into prediction groups. Others use physics-based methods to model the temperature deposition through bioheat equations and visualization softwares. Still others implement machine

learning on existing or simulated data to predict the temperature reached during treatment.

One statistical method to predict the temperature reached during a FUS treatment is the Beam index [6]. This article uses the Kranion visualizer to develop and use the model. The Beam index was created to capture the energy penetration of the focused ultrasound treatment. To do so, it calculates the ratio of energy transmitted through different parts of the skull and brain. These ratios are multiplied together and scaled to output the Beam index. This index can then be used in a temperature prediction model. This model developed is a linear-mixed-effects model. This article uses 22 patients and 163 sonications. The Beam index predicts a temperature within 3.8℃ of the true temperature 75% of the time.

Other temperature prediction models are simulation-based, used to model the physics of temperature deposition during treatment. These model types often utilize the Pennes Bio-Heat Transfer Equation. One model of this type develops a computational model and compares it to an in vitro experiment, using a simulated power field [10]. Another model is also a computational model that aims to mimic the temperature spread of the focused ultrasound treatment, using the bioheat equation, but improving on it in MATLAB [11]. Both models are examples of predicting the temperature but are simulation-based and do not rely on recorded patient data.

Lastly, a machine-learning temperature prediction method analyzes different approaches to estimate maximum pressure, power deposition, and temperature rise [12]. One model of this type tests on 19000 simulated data points, not human patient data. Overall, the random forest regression model was the most accurate. Random forest models can have their performance parameters tuned, which increases their performance. The simulation data is specific to a transducer, and the authors suggest that their model does not perform well outside of this dataset.

These existing statistical, physical, and machine learning models represent valuable work done to predict temperature in focused ultrasound treatment, a definitive measure of the treatment's success. However, these current methods lack accuracy, are too complex for daily clinical use, or lack human testing data. This project aims to use human patient data to build a machine-learning model, building on the previous model methods, to increase the accuracy and simplification of temperature prediction for FUS treatments.

Results

Temperature Predictive Model

Six models, trained on three feature subsets, were tested to determine an optimal model for temperature prediction. The models tested were: linear regression, support vector regression (SVR), random forest regression (RF), and gradient-boosted random forest (XGBoost). For the SVR and the RF models, two versions were tested. The first version used default hyperparameters, chosen from the scikit-learn function documentation, and the adjusted version used optimized hyperparameters. Three subsets of features were tested, ranging from 9-18 features. The 9 feature group resulted from the empirical feature selection. The 11 feature group was based on intuitive biological relevance. The 18 feature group contains all features, a test to ensure that removing features would not compromise accuracy. For each model, the R^2 and maximum error were reported, shown in **Figure 1**.

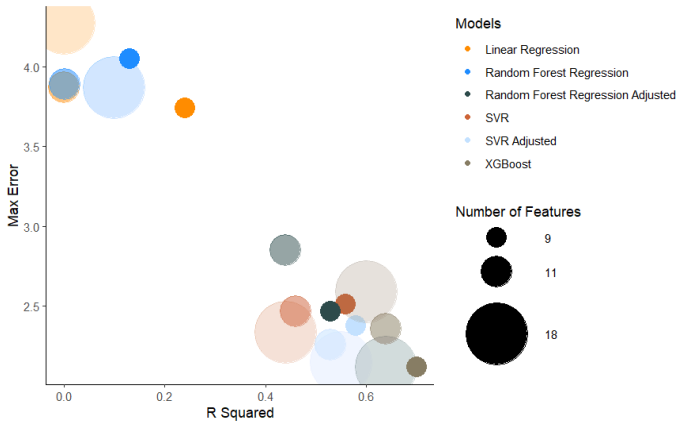


Figure 1, Model Comparison: R Squared, Error, Feature Numbers

The XGBoost model type performs best. This model is similar to the random forest regressor, which is composed of decision trees. XGBoost is a machine learning method that trains multiple gradient-boosted decision trees, weights most important trees, and combines the tree predictions to minimize final prediction error [13]**.** The use of the 9 empirically selected feature subset in the XGBoost model also minimized the lowest error. The 9 features required as model input are: sum of input energy for the treatment, focal RAS-R, focal RAS-A, focal RAS-S, mean incident beam angle, dose area, mean SDR, mean skull thickness, and number of ablative sonications.

Model Hyperparameter Tuning

The quantification of the performance in **Figure 1** was performed on a 90-10% testing/training data split. This training-favored split is necessary for machine

learning on small datasets. However, it reduced the test data to fewer than 10 patients, limiting result confidence. To increase the model confidence, leave-one-out cross-validation was used. Leave-one-out cross validation trains a model on all but one patient and then creates a prediction for the left-out one patient. The process is repeated until each patient has been tested, creating 75 temperature predictions.

The baseline model of XGBoost is often viable, but tuning its hyperparameters can improve its performance [14]. These hyperparameters control the loss function of the model, as well as the depth and size of the trees. The model hyperparameters were tuned with two steps: random search cross-validation and intuition of hyperparameter meaning. First, random search cross-validation was used to iterate through a range of hyperparameters and output the most performance-optimizing set. Because the leave-one-out cross-validation involves creating a new model for each prediction, the hyperparameter tuning must be redone for each prediction, increasing the computational cost of the model. To reduce the computational cost, for each fold, the best hyperparameters were recorded. The hyperparameters that were most often identified and intuitively appropriate were selected to be used to create one final model for all future predictions.

Model Performance and Interpretability

A direct measure to understand the performance of the model is a representative decision tree from the XGBoost forest, as shown in **Figure 2**. Decision trees display the model decision-making, especially useful for explaining the model to non-technical audiences. A decision tree showcases model decision-making through leaf values, which represent the prediction of that tree for that data point. The values of all the leaf nodes are used to form the final model prediction in conjunction with the leaf node values from the other trees in the forest.

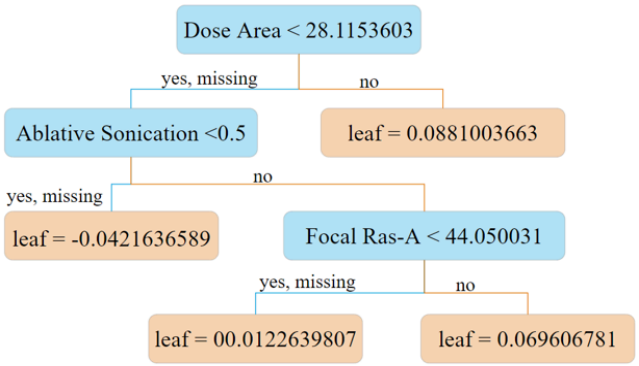


Figure 2, Representative Decision Tree from XGBoost Model

To holistically assess the model's performance, the temperature predictions of the model were plotted against the true achieved temperatures, as seen in **Figure 3**. For reference, two lines were added to the figure within ±2℃ of the true temperature. 66 predictions (88%) fall within ± 2℃ of the true temperature. 55 predictions (73%) fall within ±1℃ of the true temperature. The model had a root mean squared error of 1.34℃.

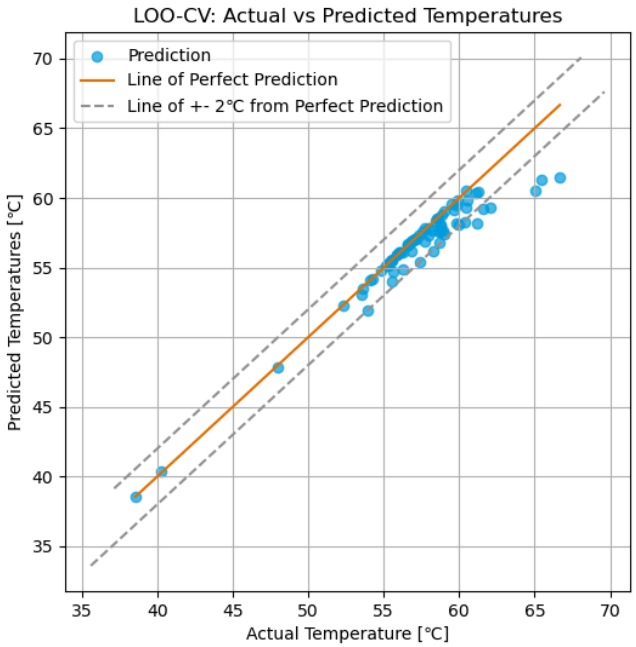


Figure 3, Model Predictions of Final Model

An ideal model has low bias and low variance, predicting accurate and consistent values. Low bias can be detected if a histogram of residuals is centered and maximal at zero, indicating it is likely to minimally predict below and above the true values. A model has low variance if the spread of residuals is narrow, indicating a consistent prediction. Our model, as seen in **Figure 4**, has a shift in residual values to the right of zero. While the peak at zero shows that our model can often accurately predict the temperatures, the shift of the curve to the right indicates there is a systemic bias in our model in which the predictions are underestimating the true values. In addition, the presence of positive residuals indicates there is variance in our model. However, the lack of large residuals outside of zero suggests that the variance is low compared to the bias.

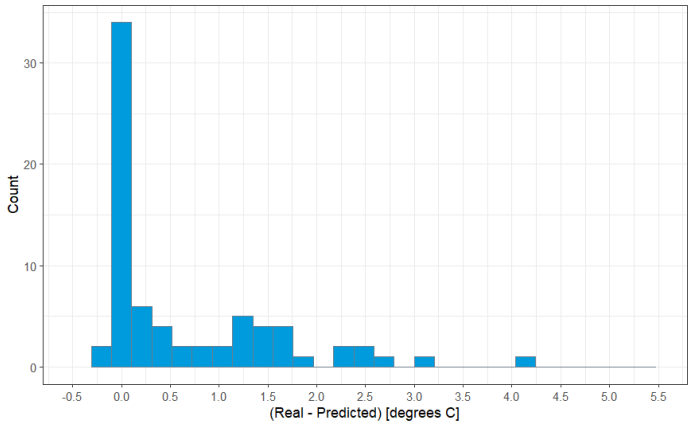


Figure 4, Histogram of Residuals from Final Model

Bias and variance can be altered through hyperparameter tuning, and one common method is to increase the number of trees. **Figure 5** shows that increasing the number of estimators or trees in our XGBoost model decreases the bias and variance. This is because each additional tree attempts to model and correct the errors made by the previous trees. However, the model reaches a stable point at around 70 estimators where additional trees do not result in decreased bias or variance and instead add computational complexity. This is the lower limit of bias and variance within the limitations of our model.

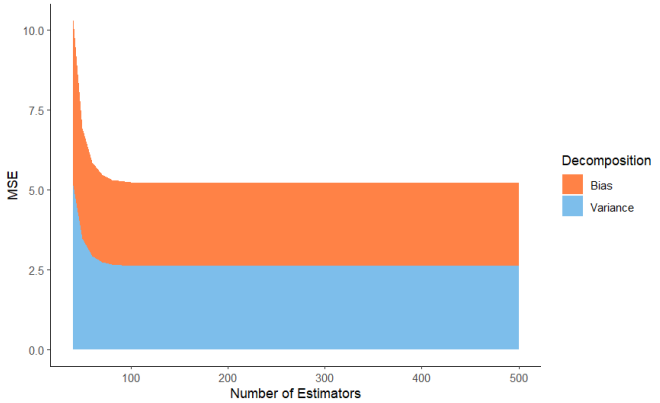
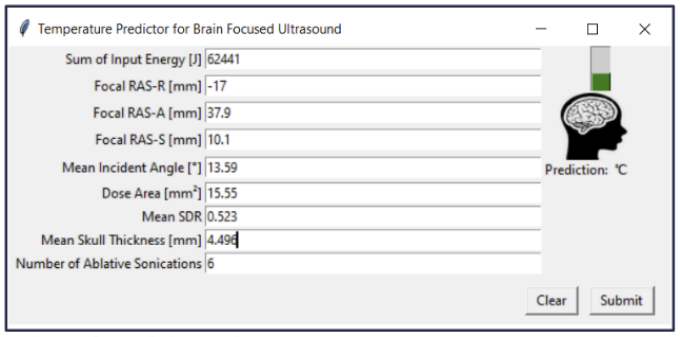


Figure 5, Bias and Variance Decomposition with Varying Number of Estimators

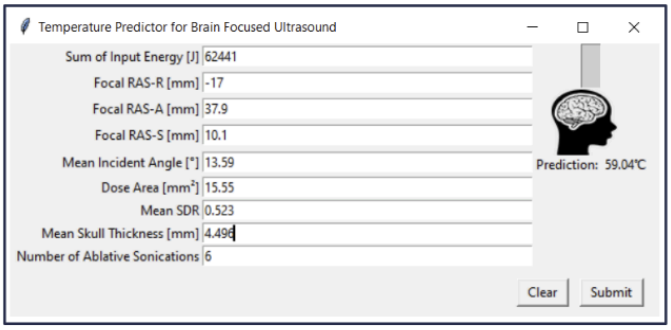
Graphical User Interface

A Graphical User Interface (GUI) was developed for users to interact with the temperature predictive model. Users can input numerical values for the 9 patient and transducer features used in the model and receive the model's prediction for the maximum temperature the individual patient's target region will reach.

The brain picture shows a rudimentary focused ultrasound patient setup to make the use of the GUI clear. While making the temperature prediction, the green progress bar mimics the downward progression of a focused ultrasound beam to make interacting with the GUI engaging and easily show the GUI is working.



a) Display While Prediction Is Running



b) Display After Prediction Is Complete

Figure 6, GUI While Running (a) and After Prediction (b)

Discussion

Our model provides a much-needed ability to accurately predict the temperature a target region will reach before FUS brain treatments. The model improves upon previous work as it provides the ability to make predictions rather than rely on statistical categorizations, it is more accurate than previous methods such as SDR and the Beam Index, it is trained on real patient data, and it is simple and accessible for further research and clinical use.

The accuracy of the model is a critical goal. As a RF model that has previously been found to perform well for FUS temperature prediction, and XGBoost is a boosted version of a RF, XGBoost inherently increases previous performance. As demonstrated through its high R^2, the final model captures important sources of variance that influence measured temperatures in this dataset. The model also demonstrates a high accuracy between predicted and true temperatures through low root mean squared error and residuals.

The accuracy of the temperature prediction shown in the GUI for future inputs relies on good faith input parameters within a reasonable biological range. For

example, an input involving the parameters of 4 ablative sonications but only 1000 J of summed energy, which does not reflect a reasonable amount of energy to achieve 4 ablative sonications, may produce a misleadingly high-temperature prediction result. To allow the most flexibility for researchers in this development stage, the model does have biological common sense input rejections. Adding warnings or rejections of biologically impossible inputs can be added in future GUI development to guide the user in providing valid inputs, especially in a clinical setting. For reference, realistic input biological values we found in the patients are shown in **Supplementary Table 1**.

In the research stage, the XGBoost model is especially useful for its flexibility. It has many hyperparameters that can be adapted, and is very suitable for large datasets, offering flexibility for future dataset expansion. The current model averages or sums certain parameters to collapse into an accessible by-patient dataset. However, there is more granular data, such as Kranion's rich by-channel estimates of mean skull thickness and SDR, available that could be used for a future model iteration. Collecting the transducer information by channel could allow more analysis of variation in skull density and skull thickness as well as energies emitted in each beam, possibly further increasing temperature prediction accuracy. Additionally, systemic model changes such as additional training and testing data could reduce bias and variance in the current model.

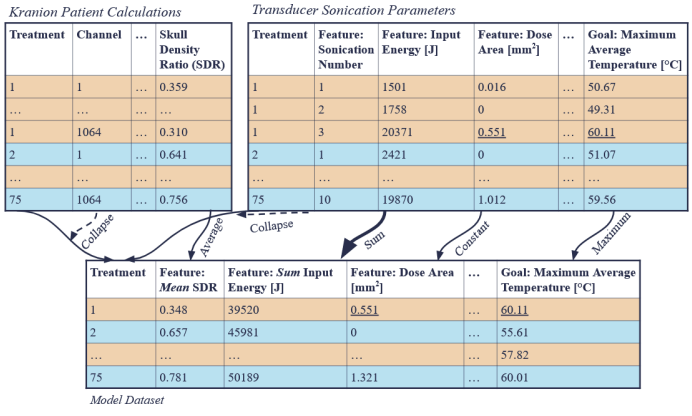
Though FUS confers high therapeutic value via its incisionless approach and has been FDA-approved for brain diseases like essential tremor for 8 years, the adoption of the technology has been slowed. One cause of slowed FUS adoption is economic barriers such as low reimbursement rates for patients [15]. The development of our model in Python, a free-to-use language, was a conscious choice to reduce economic barriers in the research stage of focused ultrasound. The packages used within Python, especially the GUI package, were chosen to be hardware lightweight and cross-platform. Additionally, input features were selected to be few and clinically reasonable to obtain. For example, the RAS inputs are target position inputs in 3 dimensions, already obtained from the MRI pre-treatment. Similarly, the desired dose area can be obtained from the MRI pre-treatment. Parameters like the mean incident angle, mean SDR, and mean skull thickness can be exported from Kranion. Only the sum of input energy and and number of ablative sonications require intuitive input and experimentation. The development of a graphical interface is crucial to producing user-accessible research, allowing focused ultrasound technology to be used clinically despite high levels of complexity. The decision tree and the GUI allow physicians to explain and interact with the machine learning model, bridging the gap between developers and clinicians. The prediction accuracy and accessibility advances of our model are designed to remove technical and economic barriers for FUS technology, allowing more patients to benefit from an incisionless brain treatment.

Materials and Methods

Patient Data Processing

94 anonymized treatments performed from 2011 to 2019 were provided by the UVA Focused Ultrasound Center. These treatments were loaded into the Kranion visualization software and patient anatomical calculations were exported [6]. Of the 94 treatments, 75 contained both a usable transducer and anatomical export file and were used for the model. Transducer and anatomical files were combined as shown in **Figure 7**. The transducer files were given by-sonication and the anatomical files were given by-beam-channel, so collapses of the sonications and channels were performed to produce by-patient input for the model.

For these collapses, we identified the maximum average temperature as the goal for prediction and held the transducer features of the dose area constant at that temperature. Other transducer features, such as the input energy and the number of ablative sonications, were summed from all previous sonications to represent the treatment progression before that goal temperature was reached. By-channel anatomical features, such as the skull density ratio and the skull thickness, were filtered to remove channels with missing data and then averaged across all remaining channels to produce a by-patient value.



Treatment	Channel	…	Skull Density Ratio (SDR)
1	1	…	0.359
…	…	…	…
1	1064	…	0.310
2	1	…	0.641
…	…	…	…
75	1064	…	0.756

Treatment	Feature: Sonication Number	Feature: Input Energy [J]	Feature: Dose Area [mm^2]	…	Goal: Maximum Average Temperature [°C]
1	1	1501	0.016	…	50.67
1	2	1758	0	…	49.31
1	3	20371	0.551	…	60.11
2	1	2421	0	…	51.07
…	…	…	…	…	…
75	10	19870	1.012	…	59.56

Treatment	Feature: *Mean* SDR	Feature: *Sum* Input Energy [J]	Feature: Dose Area [mm^2]	…	Goal: Maximum Average Temperature [°C]
1	0.348	39520	0.551	…	60.11
2	0.657	45981	0	…	55.61
…	…	…	…	…	57.82
75	0.781	50189	1.321	…	60.01

Figure 7, Data Structure, Collating, and Collapsing Data by Patient

Patient and Transducer Feature Selection

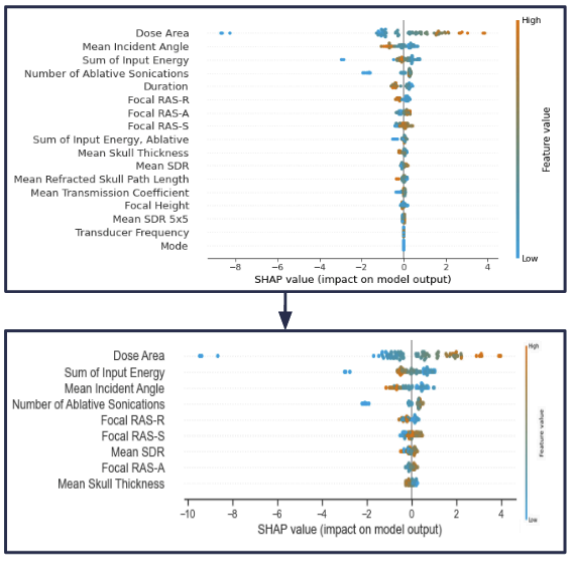
From the processed data, certain anatomical and mechanical features of the treatment were selected, and the model was trained on this subset. The feature subset was chosen using the intersection of features identified as optimal by two methods: SHapley Additive exPlanations (SHAP) value calculations for each feature and feature correlations.

SHAP values, based on game theory, predict the probability of the feature contributing to the model's output [16]. We utilized the SHAP package in Python to calculate these values. A SHAP value is calculated for each feature for each datapoint, and features are ranked in order of mean absolute SHAP value to form a feature importance plot, as in **Figure 8a**.

Features with a high numerical value are colored orange and features with a low numerical value are colored blue, adding additional context to the feature importance rankings. A positive SHAP value means the feature datapoint raises the model's final temperature prediction, while a negative SHAP value means the feature datapoint lowers the model's final temperature prediction.

The correlations for all features were utilized by plotting each feature against all other features using the seaborn package in Python, as in **Figure 8b**. Features that were observed to exhibit a correlated relationship with another feature were removed. When two features were correlated, the feature with the lower SHAP value was removed; hence, the SHAP value calculations and feature correlations were used iteratively.

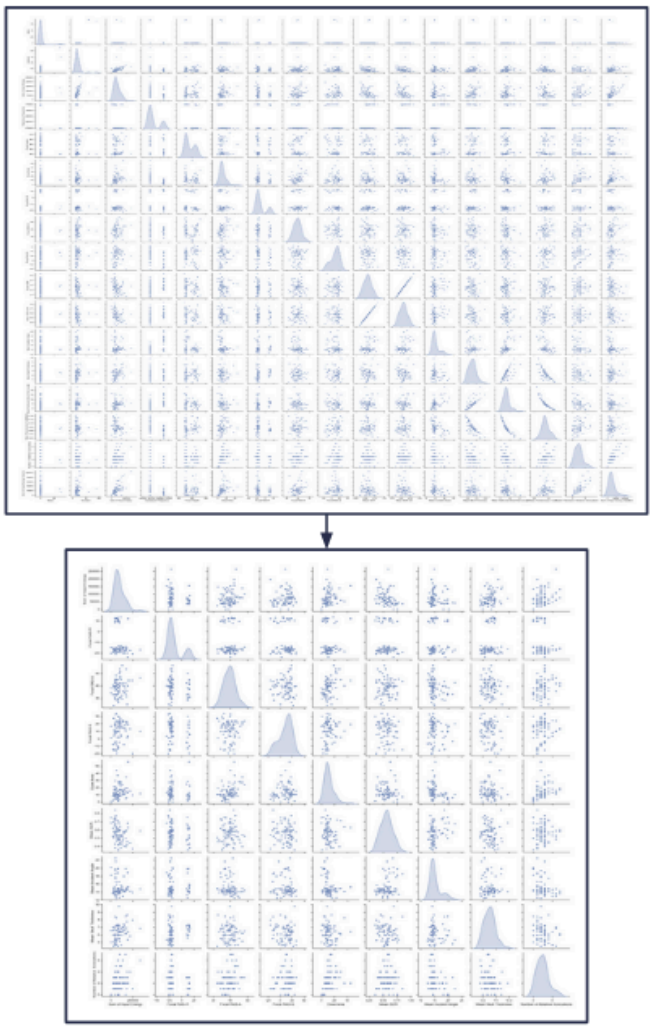
>17 Initial Features



9 Final Features

a) Removing Minimally Contributing Features via SHAP

>17 Initial Features



9 Final Features

b) Removing Redundant Features via Correlation

Figure 8, Model and Feature Selection: Removing Redundant and Minimally Contributing Features

Finally, any remaining features with low SHAP values were removed. The final features were 9 high-SHAP, low-correlation features. The number of features was minimized as much as possible without compromising model accuracy to reduce user burden.

Model Selection

To select the best models, six different models were trained from Python's scikit-learn package [17]. For this model selection, the data is split in a 90/10 split using train_test_split() in Python. The first one is linear regression, using the LinearRegression() function. The second one, SVR, uses the sklearn.svm package. Two of the parameters are changed for the second test of the SVR, gamma and C. Gamma is set to 'auto', which sets the

coefficient for the function used to train the model. C is the regularization parameter, which we set to 10. The next model is the random forest regressor, which is trained using the RandomForestRegressor() function in Python. Some of the changes to the hyperparameters include: changing the depth of the trees, changing the tree splits, and changing the number of estimators. The last model is the XGBoost model, it uses a similar method as the random forest and uses the XGBoostRegressor() function.

The results were plotted in R, using the ggplot2 package [18]. The R^2 and error are on the axis and using scale_size_manual(), the sizes of the data points were changed based on the feature number. The opacity was also adjusted with the smaller feature numbers being more opaque, to make them stand out more.

At this point the model is not tuned. For the final version of the model, leave one out cross-validation and hyperparameter tuning was used. This was done using the LeaveOneOut package in Python which allows for a new train_test_split to be created for every fold. This means that a loop is created with one test data point taken out each time, for each iteration, the prediction is outputted. The output is then used to test the performance, as seen in the Results section.

Model Hyperparameter Tuning

Hyperparameter tuning was first done using the RandomSearchCV function from the scikit-learn package in python. Hyperparameter tuning was run for each fold, and for each fold, the optimal set of hyperparameters are returned. The ranges tested for the hyperparameter were chosen from the documentation. After using the most commonly outputted ones, the final parameters are shown in **Supplementary Table 2**.

Model Performance Evaluation

To output the decision tree the best iteration from the XGBoost model is outputted and plotted with plot_tree. The model retains the information of each tree, and the best iteration is outputted (using xgb_model.get_booster().best_iteration()), to show a more comprehensive tree. Some trees are very small with just one or two decisions, which is not intuitive for physicians, this is why the best iteration tree is outputted.

To visualize the predictions of the model and get the R^2 outputted, the predictions and real values are plotted using the matplotlib package [19]. After each iteration, the test values extracted from the leave-one-out method are used to predict the temperature. For each fold, a new prediction is stored in a list. Using the r2_score function from the scikit-learn package, the R^2 is calculated between the real and predicted values. On the plot, a y=x line is plotted to visualize the perfect predictions, which helps visualize how far from the real values the predictions are.

To plot the residuals to visualize the variance, the real and predicted values were extracted from python and imported into R. First, the predicted values were subtracted from the real values in Excel and uploaded into R using the read_excel function, native to R. Then, using ggplot2, a histogram was created to help the visualization. The x-axis was adjusted to demonstrate more values. To visualize the bias and variance decomposition, the bias_variance_decomp() function was used, outputting the bias and variance with varying values of estimators. The results are aggregated and imported into R.

GUI Development

The GUI was developed in Python using the preloaded tkinter package. The tkinter package is designed to be lightweight and cross-platform, facilitating compatibility with many computer systems [20]. The GUI imports the trained model within the same script that creates the interface, creating ease of use by not requiring users to first run any other files to train the model. The brain picture was created by saving a transparent png. The brain beam was created using a progressbar widget from the ttk package within tkinter. The progressbar was turned vertically and the direction reversed by starting the bar at full and decreasing the bar value using the time package within Python.

End Matter

Author Contributions and Notes

The authors declare no conflict of interest.

Acknowledgments

We thank the UVA Focused Ultrasound Center for providing their patient data and the Focused Ultrasound Foundation for their lasting support. We are deeply grateful to Samarth Singh for his data science advice.

References

[1] Y. Meng, K. Hynynen, and N. Lipsman, “Applications of focused ultrasound in the brain: from thermoablation to drug delivery,” *Nat. Rev. Neurol.*, vol. 17, no. 1, Art. no. 1, Jan. 2021, doi: 10.1038/s41582-020-00418-z.

[2] “Premarket Approval (PMA).” Accessed: May 02, 2024. [Online]. Available:

https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfPMA/pma.cfm

[3] "A Review of the Current Therapies, Challenges, and Future Directions of Transcranial Focused Ultrasound Technology: Advances in Diagnosis and Treatment | Neuro-oncology | JAMA Neurology | JAMA Network." Accessed: May 02, 2024. [Online]. Available: https://jamanetwork.com/journals/jamaneurology/article-abstract/2664950

[4] "Focused Ultrasound for Brain Diseases: A Review of Current Applications and Future Perspectives - ClinicalKey." Accessed: May 02, 2024. [Online]. Available: https://www.clinicalkey.com/#!/content/playContent/1-s2.0-S1959031823000398?returnurl=null&referrer=null

[5] "Kranion," Focused Ultrasound Foundation. Accessed: May 02, 2024. [Online]. Available: https://www.fusfoundation.org/for-researchers/resources/kranion/

[6] F. Sammartino, D. W. Beam, J. Snell, and V. Krishna, "Kranion, an open-source environment for planning transcranial focused ultrasound surgery: technical note," *J. Neurosurg.*, vol. 132, no. 4, pp. 1249–1255, Mar. 2019, doi: 10.3171/2018.11.JNS181995.

[7] "Skull density ratio and treatment outcomes in essential tremor patients treated with transcranial MR-guided focused ultrasound thalamotomy," Congress of Neurological Surgeons. Accessed: May 02, 2024. [Online]. Available: https://www.cns.org/meetings/archived-abstracts-detail/skull-density-ratio-and-treatment-outcomes-in-essential-tremor-patients-treated-with-tran

[8] "Impact of skull density ratio on efficacy and safety of magnetic resonance–guided focused ultrasound treatment of essential tremor in: Journal of Neurosurgery Volume 132 Issue 5 (2019) Journals." Accessed: May 02, 2024. [Online]. Available: https://thejns.org/view/journals/j-neurosurg/132/5/article-p1392.xml?tab_body=fulltext

[9] "CEM43°C thermal dose thresholds: a potential guide for magnetic resonance radiofrequency exposure levels? - PMC." Accessed: May 02, 2024. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3799975/

[10] V. Hadjisavvas and C. Damianou, "A simulation model for predicting the temperature during the application of MR‐guided focused ultrasound for stroke treatment using pulsed ultrasound," *AIP Conf. Proc.*, vol. 1359, no. 1, pp. 253–258, Sep. 2011, doi: 10.1063/1.3607914.

[11] P. Revathy and V. Sadasivam, "Computational Model of Pixel Wise Temperature Prediction for Liver Tumor by High Intensity Focused Ultrasound Ablations," in *Information Systems Design and Intelligent Applications*, S. C. Satapathy, J. K. Mandal, S. K. Udgata, and V. Bhateja, Eds., New Delhi: Springer India, 2016, pp. 137–146. doi: 10.1007/978-81-322-2752-6_13.

[12] "Bioengineering | Free Full-Text | Modeling Focused-Ultrasound Response for Non-Invasive Treatment Using Machine Learning." Accessed: May 02, 2024. [Online]. Available: https://www.mdpi.com/2306-5354/8/6/74

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in KDD '16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[14] "XGBoost Parameters Tuning | Complete Guide With Python Codes." Accessed: May 02, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

[15] I. Pérez-Neri, A. González-Aguilar, H. Sandoval, C. Pineda, and C. Ríos, "Potential Goals, Challenges, and Safety of Focused Ultrasound Application for Central Nervous System Disorders," *Curr. Neuropharmacol.*, vol. 20, no. 10, pp. 1807–1810, Aug. 2022, doi: 10.2174/1570159X20666220201092908.

[16] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions." arXiv, Nov. 24, 2017. doi: 10.48550/arXiv.1705.07874.

[17] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.

[18] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*, 2nd ed. 2016. in Use R! Cham: Springer International Publishing : Imprint: Springer, 2016. doi: 10.1007/978-3-319-24277-4.

[19] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, May 2007, doi: 10.1109/MCSE.2007.55.

[20] F. Lundh, "An Introduction to Tkinter"

Supplementary Information

Table 1. Biological Common Sense Parameter Ranges from Model Dataset

Input Parameter	Units	Minimum	Maximum
Sum of Input Energy	J	500	263535
Focal RAS-R	mm	-24.04	13.3
Focal RAS-A	mm	32.06	48.2
Focal RAS-S	mm	-20	33.6
Mean Incident Angle	°	12.49	22.46
Dose Area	mm^2	0	56.23
Mean SDR	–	0.36	0.84
Mean Skull Thickness	mm	4.44	9.79
Number of Ablative Sonications	–	0	7

Table 2. Hyperparameters for Final XGBoost Model

Parameter	Value
Objective	Squared Error
Column Sample by Tree	0.8
Learning Rate	0.1
Maximum Depth	3
Minimum Child Weight	3
Number of estimators	500
Subsample	0.8
Early Stopping Rounds	5