VR Tracker Location and Rotation Predictions using HTC Vive Tracking System and Gradient Boosting Regressor

Lin Hall, Ping Wang, Grayson Blankenship, Emmanuel Zenil Lopez,
Chris Castro, Zhen Zhu, Rui Wu
East Carolina University, Greenville, NC USA

Abstract

Machine learning and virtual reality technologies have become focal points for research and development in recent years. In this study, we propose a framework to estimate and visualize the position and rotation of human spines, based on predictions made with machine learning. HTC Vive trackers are used to simulate the bone structure. Truth reference data for position and rotation are collected in the HTC system, which are used to evaluate the performance of our solution. Preliminary results show that the pose of the simulated structure can be accurately predicted. The proposed framework can be beneficial to medical training and surgical operations.

Key Words: Tracking, virtual reality, machine learning, gradient boosting regressor

1 Introduction

Rapid advances in technology and medical device development in the 21st century are bringing about a new era of medicine, contributing to healthier and more productive lives. As technology and patient complexity continues to increase, demands for novel approaches to ensure competency have arisen [17].

Machine learning as a means of pattern recognition has been heavily utilized over the last decade. By definition, machine learning is pattern recognition that explains the surrounding environment. Further, this pattern recognition is achieved by modeling human intelligence [7]. In this paper, we utilize machine learning techniques to predict the motion of biomedical systems, such as human spines. Users can visualize the location and orientation of human spines in a virtual reality environment in real time. It provides virtual reality experiences to medical professionals during training, or to serve as an additional visualization and guidance tool during actual surgical operations.

Existing machine learning studies in medical research have been largely focused on clinical datasets and patient diagnoses [3, 15], although some predictive analyses have been attempted in certain areas, including: cancer [4, 14], stroke [11], and dementia [1]. In this work, we explore the real-time prediction capability of machine learning, which is a critical component in image-guided surgery.

During surgical operations, the patient may not be completely motionless. The human body may experience small changes in position and/or orientation. Most existing visualization tools of the human spine rely on mathematical models and imaging equipment [10], which are not capable of handling real-time motion. By contrast, in improved spinal visualization systems [23], machine learning has been successfully used to image the human spine, with quality even rivaling that of manual imaging [9].

On the other hand, virtual reality (VR) is a relatively new concept in medical research. Typically, VR has been considered an educational tool [20]. Virtual reality simulator becomes a powerful tool for surgical trainees to repeatedly practice without potential harm to patients and animals. Traditionally, VR was not widely used in high-fidelity applications. However, recent development in VR technology has enabled higher-accuracy solutions. A 2021 study found that using augmented and virtual reality resulted in 97% accuracy for pedicle screw placement [8]. Unfortunately, this study relied on a static spine, which does not model patient motion during surgery. For example, pedicle screw placement can cause shifts along the spine during a surgical procedure[8]. Therefore, we propose to utilize machine learning techniques to account for such spinal motion.

One of the main applications of machine learning is data visualization [6], particularly so for medical research. The importance of data visualization in terms of knowledge extraction has been documented [21]. Further, when combined with VR, visualization results in a complete immersion into the data [6]. In the proposed system, visualization will be implemented as a three-dimensional immersive VR experience. A section of human spine will be simulated with multiple HTC Vive Pro trackers in this work. Trackers are used to simulate motion of a spine section that is partially rigid. The non-rigid motion of the spine will be predicted and verified with the HTC system. The trackers are visualized using the Unity Virtual Reality Environment in this work.

The main contribution of this paper is to provide a software/hardware framework which integrates VR with machine learning to track, predict and visualize the position and orientation of VR trackers. The framework includes prediction of time series data obtained from the simulated human spine, for which we use a gradient boosting regressor model. Furthermore, we mitigate data outliers by utilizing the extreme event split technique in order to improve the prediction functionality. Finally, the simulated human spine will be visualized in VR. Also, the framework can support other medical visualization applications.

In section 2 of this paper, previous work related to our study is reviewed. In section 3, an introduction to the software and hardware components, architecture of the proposed framework and a process workflow from data collection to virtual reality visualization are presented. The machine learning models and the extreme event split approach are discussed in section 4. Section 5 presents results of spine motion prediction, followed by conclusions.

2 Related Work

The efforts of Bissonnette to distinguish surgical training levels using virtual reality simulator and machine learning methods suggest that virtual reality and machine learning can be powerful tools for surgical training and evaluation. The authors divided spine surgeons, spine fellows, orthopaedic and neurosurgery residents, and medical students from 4 Canadian universities into two groups (senior and junior) according to their training levels. 22 participants were senior and 19 were junior. All the participants were asked to perform a spinal surgery in a virtual reality environment. The virtual hemilaminectomy required participants to remove the L3 lamina with a simulated burr in their dominant hand while controlling bleeding with a simulated suction instrument in their non-dominant hand [2].

Participants were required to remove the L3 lamina in five minutes, without damaging surrounding tissues. Their position, angle, force application of the simulated burr and suction instruments, and removed tissue volumes during the procedure were recorded at 20 ms intervals. The data were collected as metrics for training machine learning algorithms. Five classification algorithms were applied: support vector classifier, K-Nearest Neighbors classifier, Linear Discriminant analysis, Naive Bayes classifier, and decision tree classifier. Regression algorithms can also be combined with virtual reality technique in surgical field. Dubin implemented machine learning algorithms to develop regression models and to predict Global Evaluative Assessment of Robotic Skills(GEARS) score using a VR simulator[5]. GEARS is a validated surgical proficiency testing tool, which has been widely used in training programs. 74 participants were required to perform a basic VR exercise (Ring and Rail1) and a complex VR exercise (Suture Sponge1) on two simulators–dV-Trainer(dVT) and da Vinci Skills Simulator(dVSS). The simulator gave scores of each exercise for each participant. And the recorded video was sent to human subject matter experts for review using the GEARS tool. Linear regression models were generated for each exercise on each simulator to predict GEARS score based on simulator score.

Although both works combined VR with machine learning, they used relatively simple machine learning algorithms, and were focused on the performance from the simulators. Bissonnette et al built classification model using Support Vector Classifier to distinguish senior level and junior level of surgeons. Dubin built simple linear regression to predict the GEAR score of medical students. Besides, the related research built models on relatively small datasets. Bissonnette has forty-one participants for model building and Dubin built linear regression model on 74 participants. The machine learning model required in the spine motion prediction task is a little more complicated. We predict six degrees of freedom in spine motion, three positional variables and three rotational variables. The performance of prediction will be evaluated on all six variables. The input of this model includes thousands of observations. The complexity of the input data requires special handling of extreme values, or data anomalies. An extreme value is an observation at the boundaries of the domain.

Anomaly detection in time series has attracted considerable attention due to its importance in many real-world applications including intrusion detection, energy management The finance [19]. Most anomaly detection methods require manually set thresholds or assumptions on the distribution of data. Isolated forest algorithm is one of the commonly used extreme value detecting algorithms. The term isolation in this case means "separating an instance from rest of instances". The two processing stages of isolation forest include training stage and testing stage. The training stage builds isolation trees using subsamples of the training set. The testing stage passes instances in testing set to obtain anomaly score for each instance. In the training stage, isolation trees are constructed by recursively partitioning a subsample X' until all instances are isolated. Each isolation tree is constructed using a subsample X' randomly selected without replacement from X[24].

The normal points tend to be isolated at the deeper end of the tree, whereas anomalies are closer to the tree root, due to their singularity nature. The shorter the average path length, the higher the chances to be anomalies[24]. In teintervsting stage, outliers are identified and labeled based on anomaly score of each instance. The 95% quantile method is used to determine the threshold of extreme value and the instances with anomaly score greater than the threshold are classified as outliers. The extreme value machine (EVM) introduced in 2018, has become an important tool in multivariate statistics and machine learning in the past few years. Generalized Pareto distribution(GPD) classifier is an alternative approach of EVM. It requires the generalized Pareto distribution assumption from extreme value theory. The idea of the EVM is to approximate the distribution of the margin distance of each point in each class using extreme value theory. A new point is then classified as normal if it is inside the margin of some point in the training set with high probability[22]. We applied the DSPOT algorithm of splitting the data into normal events and extreme events. Then we built machine learning models on normal dataset and extreme dataset separately and compared the results with that of model built on dataset without extreme event splitting.

Another challenge in motion prediction lies in the fact that it is essentially a time-series prediction problem. The sliding window technique has been utilized to preserve the temporal relationship of the data [12]. In this technique, the size of the window is of particular importance [12, 13].

3 Methodology

In this work, HTC Vive Pro VR trackers are used to represent parts of the human spine. Two VR tracks simulate rigid-body motion the human spine, whereas a third tracker simulates motion of unknown model. Using a time series regression model, we use location and orientation of the first two trackers to accurately predict the "third" VR tracker. The actual location and orientation of all the trackers can be accurately measured by the VR system, which allows us to assess the accuracy of the machine learning model (R^2, *rmse*, etc.).

The trackers were placed on an HTC Vive Pro racket. To train the machine learning model, motion data including the X, Y, and Z position as well as pitch, roll, and yaw angles were collected using the racket. Next, the motion data used in training were reprocessed, and fed into the time series regression model. After training is completed, prediction accuracy were assessed. Extreme event split and sliding window techniques were implemented to improve the quality of machine learning. Subsequently, this output prediction was fed into a Unity application, such that all three trackers could be displayed in the Unity application scene.

3.1 Architecture

The overall system includes three main components: the VR hardware (HTC Vive Pro trackers and racket), the backend software (machine learning algorithms on a Flask server), and the frontend software (SteamVR and Unity). In this section, we will discuss how the components are connected and interfaced with each other.

Flask is a popular web application framework. In this study, a Flask server is utilized for real-time prediction of the trackers. The processed positional data is input into the server. As a result, when activated, the server will generate predicted positional coordinates of the tracker to be visualized in Unity.

SteamVR is an expansion of the Steam gaming engine that adds a virtual reality component to the gaming experience. In this study, SteamVR is utilized in the data collection and visualization of each tracker. For data collection, SteamVR is utilized to ensure the connection between the HTC Lighthouses (instruments that create the virtual reality boundary) and the HTC VR Headset and Trackers. Further, the data is collected by performing movements in each positional and rotational direction. For visualizing the trackers, SteamVR communicates with Unity to create a scene, or virtual reality environment.

Unity is a real-time development platform, typically used in gaming. However, Unity can be utilized to simulate environments. In this study, Unity is used to visualize the trackers. As mentioned before the visualizations are called scenes.

Figure 1 illustrates the visualization process.

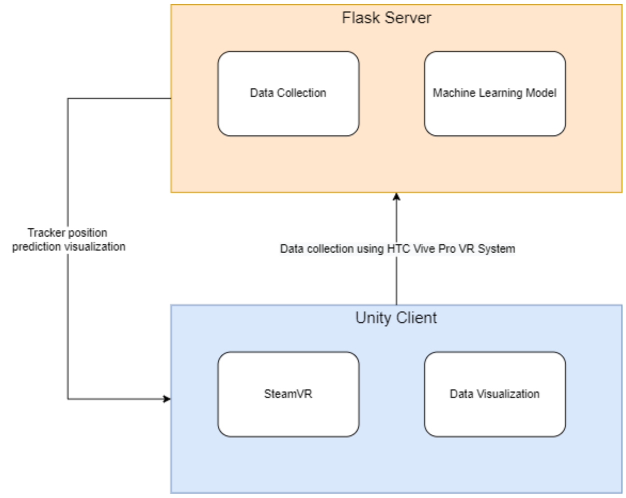


Figure 1: A flowchart illustrating the process in how data collection occurs. Here, the orange section represents components utilizing the Flask server. Likewise, the blue section represents components utilizing the Unity client. First, the data is collected from the HTC Vive Pro Setup. Next, the data is preprocessed and sent to the Flask server for the real-time tracker location prediction. The prediction is generated using a Gradient Boosting Regressor machine learning model. Each HTC Vive Pro Tracker is attached to an HTC Vive Pro Racket, which is considered a rigid body. Finally, using SteamVR and Unity, the tracker locations are visualized

3.2 Data Collection with VR Trackers

Machine learning is heavily dependent on the quality of training data. Therefore, data collection is an integral component to a successful machine learning algorithm, such as gradient boosting regressor shown in Figure 1. The HTC Vive Pro Trackers used in this work have a 270-degree field of view, which allows for data collection in virtually every direction. As afore mentioned, three trackers are manually attached to an HTC Vive Pro Wireless Racket. Each tracker has a unique identifier. The racket allows us to place trackers in a straight line with equidistant positions. On the HTC Vive Pro racket in Figure 1, the predicted tracker is in the middle, and is adjacent on either side by two other trackers. It provides a simplified model of human spine. Although the racket can only simulate linear motion of the third tracker within a rigid body, it is not a requirement for the machine learning algorithm. Non-rigid motion of the tracker can also be predicted.

Sensors attached to the trackers and a HTC Vive Pro Headset must be synced to the Steam virtual reality software, and to the HTC Vive Pro Lighthouses (Base Stations). These wireless lighthouses are responsible for determining the position of the sensors in a VR environment. Typically, these lighthouses are placed approximately six feet apart in a room. The HTC Vive Pro Headset must be active at all times for data collection to occur. Failure to do so will result in the data being inconsistent, or even uncollected.

A Steam virtual reality scene is first initialized, in order to create the required VR environment. In this environment, the raw data are recorded. Unity provides an API to calculate the location of the sensors, which is relative to the lighthouse base stations. The point of origin is selected for a spatial Cartesian XYZ system. When activating the Steam VR environment, it is imperative to have the HTC Vive Pro Racket at the point of origin in order to ensure accurate data collection. Since the sensors are rigidly attached to the trackers and headset, position of multiple sensors on a tracker/headset can then be used to estimate the position and orientation of the whole tracker/headset.

3D position and 3D orientation data (pitch, roll, and yaw angles) are collected. Position and orientation are recorded in the coordinate system defined by the virtual reality environment created using the Unity Game Engine. In the 3D spatial Cartesian coordinate system, X and Y are horizontal axes and Z is the vertical axes. However, X and Z in the Unity environment are horizontal, corresponding to the Cartesian X and Y directions respectively. The Unity Y axis is vertical, equivalent to the Cartesian Z direction.

While the Steam VR environment scene is active, the output of data occurs continuously until the scene is stopped. During each step of the data collection, we focus on only one of the dimensions. The HTC Vive Pro Racket (with each tracker attached) is shifted in the desired dimension at different positions. For example, if data collection is focused on the Unity X direction, the racket motion will be primarily on the X direction. The motion on X direction will be random, and will be repeated at various positions. At each position, data collection lasts approximately 15-20 seconds. Subsequently, a dataset is created for each of the six dimensions. Therefore, there will be six individual data files.

The recorded data are preprocessed to remove incomplete samples and null values, and subsequently recorded in comma-separated value (csv) format. The csv files are cleaned to remove redundant information. For instance, if data is collected for the X position, the Y and Z positional information is removed from this file. As a result, the data are less noisy, purely focused on one direction at a time. It allows for direct observability in each of the dimensions.

Preprocessed data files are tested for quality. Here, simple linear regression is used from the scikit-learn machine learning library [16]. To ensure the accuracy of the data collection for each directional file, the R^2 value is calculated. If the dataset was incomplete, or held any null or unaccepted data types, a modeling error is returned. Further, if the R^2 value was low, the data file will be recollected.

3.3 Extreme Event Split

We applied Drift Streaming Peaks-Over-Threshold(DSPOT) to detect extreme events of the time series data and split the dataset into normal dataset and extreme dataset. As stated earlier, isolation forest extreme value detector and GPD classifier rely on either manually set thresholds or assumptions on the distribution of data. By using the DSPOT approach, we do not assume the distribution of the value but rely on extreme value theory to estimate accurately low probability areas and then discriminate outliers [19].

3.4 Real-time Prediction

The preprocessed data is sent to a Flask Server. On this server, a gradient boosting regressor machine learning model is implemented. The predicted location and orientation of the tracker are updated and distributed continuously as long as the server is active. The prediction is then sent to the Unity application and are visualized. Figure 2 shows the visualization of the trackers, including the predicted location.

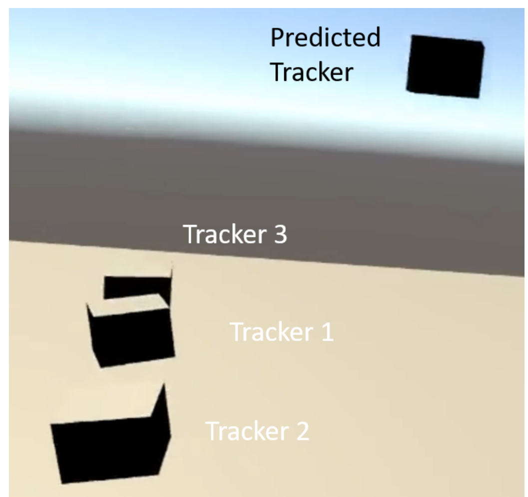


Figure 2: An illustration of the virtual reality visualization of the HTC Vive Pro Wireless Racket setup with the attached trackers. During surgery, sensors are placed on the spine, trackers 2 and 3 are external sensors, and tracker 1 is for bone location. When predicted correctly, the visualized tracker should overlay tracker 1

3.5 Window Slider Technique

Instead of predicting the target variable using the whole training dataset, the window slider technique helps improve the accuracy of predictive models by capturing the most complete information possible from the dataset.

Figure 3 [18]shows how the sliding window technique reshapes the information by windows with fixed size. The X axes shows the time, and the y axes shows the response variable. Predictors are not shown in this figure. The window size of this example is 4, which means that the model is going to map the 4 observations in this window and predict the value at time

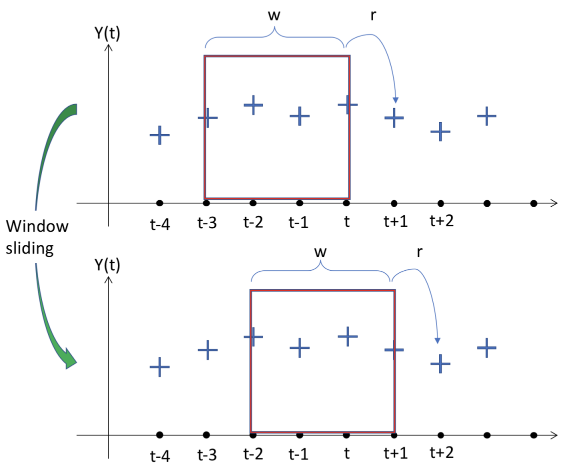


Figure 3: Window slider

t+1. Then the sliding window moves forward one time step and a response variable at t+2 is predicted. The sliding window continues moving forward and proceeding the same prediction step until the end of the time series dataset.

Instead of predicting one variable in a time series dataset as shown above, the tracker dataset we applied sliding window technique to has six predictor variables–rotational X,Y,Z variables of tracker one and two, and one response variable–rotational X variable. We have n-w windows in total, where n representing the sample size of training set, and w representing the window size. Sliding window technique was applied to machine learning models and RMSE values will be calculated to evaluate the accuracy of models.

Figure 4 shows an example of defining windows in the tracker dataset with window size set to 3. The data records in the black frame are a training set used to train the models. There are seven predictor variables in this training set–Δt and X1-X6, and one response variable–Y. The response variable next to the window is predicted with the trained model. This procedure is repeated n-w times as the window slides down one row each time.

Window

Δt	X_1(t)	X_2(t)	X_3(t)	X_4(t)	X_5(t)	X_6(t)	Y(t)
0	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-
…	-	-	-	-	-	-	-
n	-	-	-	-	-	-	-

Predict

Figure 4: Window with size=3

Figure 5 shows the new dataset generated using the sliding window technique with window size set to 3. Models are trained using this new dataset and RMSE values are calculated to evaluate the model. This procedure is repeated with different sample sizes.

To determine the window size, autocorrelation is utilized. Autocorrelation represents the degree of similarity between a time series by measuring the relationship between a variable's current values and its historical values over successive time intervals[25]. Eq. 1 is used to calculate the autocorrelation.

#	Δt(0)	Δt(1)	Δt(2)	X1(0)	X2(0)	X3(0)	X4(0)	X5(0)	X6(0)	X1(1)	…	X6(2)	Δt(3)	Y(3)
0	0	-	-	-	-	-	-	-	-	-	…	-	-	-
1	0	-	-	-	-	-	-	-	-	-	…	-	-	-
2	0	-	-	-	-	-	-	-	-	-	…	-	-	-
…	0	-	-	-	-	-	-	-	-	-	…	-	-	-
n-w	0	-	-	-	-	-	-	-	-	-	…	-	-	-

Figure 5: New dataset

$$\hat{\rho}_k = \frac{\Sigma_{t=k+1}^{T}(r_t - \bar{r})(r_{t-k} - \bar{r})}{\Sigma_{t=1}^{T}(r_t - \bar{r})^2} \tag{1}$$

4 Results

As stated earlier, the gradient boosting regressor model, along with the extreme event split and the sliding window technique, were utilized in this study. Table 1 shows the results of each model run for each position and rotation.

4.1 Gradient Boosting Regressor

Initial results of the gradient boosting regressor model show a high accuracy with low root mean square error for each position (X, Y, Z) and pitch rotation. However, roll and yaw did not produce low-error results. This high accuracy is high due to the linearity, as well as continuity of the data. The lower accuracies in roll and yaw can be attributed to the non-linearity of the data, where a pitch movement is more linear than roll or yaw movements. Table 1 shows the accuracy of the predicted tracker locations for each position and rotation in relation to the actual tracker coordinates. As a whole, the GBR performed well.

The extreme event split and sliding window technique performed better than the base gradient boosting regressor model. This indicates the data contained several outliers, and that the base model does not perform as well when the data is inspected all at once. Also, this is important since the data collection is performed by manual simultaneous movements of the trackers. Therefore, it is possible not all of the movements are consistent.

Consequently, the better comparison is between the extreme event split and the sliding window technique. The extreme event split performed better overall for positional movements, while the sliding window technique performed better for rotational movements. This is a result of the extreme event split accounting for the outliers in the mostly linear positional dataset. However, the sliding window technique performs better with the rotational data due to its multiple examinations of the dataset, which are correctly considered windows. Further, the rotational data is the least linear of the collected data.

Table 1: This table displays the model results of the base gradient boosting regressor model prediction, the extreme-event split, and the sliding window technique on the input data. Root Mean Square Error was calculated for each type of model run. The GBR model performed well for each position and rotation, with the exception of the pitch rotation. The extreme event split was better for the X, Y, and Z positions, while the sliding window technique performed best for the pitch, roll, and yaw rotations

Method	Metric	Position					
		X	Y	Z	Pitch	Roll	Yaw
GBR (Base Data)	RMSE	0.100	0.117	0.204	0.444	848.046	515.981
GBR (Extreme Event Split)	RMSE	0.036	0.036	0.036	0.353	18.266	11.673
GBR (Sliding Window)	RMSE	0.313	0.260	0.313	0.250	14.312	9.366

5 Conclusion

In this paper, we proposed a framework to visualize and predict HTC Vive trackers. The experimental results show that our proposed method is promising and can be possibly applied to medical usage. Here, the extreme event split proved best for improving the model results for the X, Y, and Z positions. Conversely, the sliding window technique performed best for the pitch, roll, and yaw rotations.

In Table 1, we have shown the results of the predicted tracker location in relation to tracker 1 for this research. As a result, the gradient boosting regressor model has proven useful for the machine learning application of this research, as well as the utilization of the extreme-event split and sliding window techniques.

With that said, some improvements can be made to ensure further decreases in error within the models. Parameter tuning of the extreme-event split and sliding window technique would further improve accuracy within the dataset. Also, the overarching goal of this research is to visualize tracker locations. By utilizing a mixture of the techniques, a better visualization would be possible. More specifically, utilizing the extreme event split for the positional data, and sliding window technique for the rotational data should produce a more accurate visualization.

Future research goals include utilizing multiple techniques, as mentioned above. Also, prediction for a non-rigid body is significant for an actual human body.

6 Acknowledgements

This work is supported in part by the National Science Foundation IUSE/PFE:RED award #1730568.

References

[1] Gopi Battineni, Nalini Chintalapudi, and Francesco Amenta. Machine Learning In Medicine: Performance Calculation of Dementia Prediction by Support Vector Machines (SVM). *Informatics in Medicine Unlocked*, 16:100200, 2019.

[2] Vincent Bissonnette, Nykan Mirchi, Nicole Ledwos, Ghusn Alsidieri, Alexander Winkler-Schwartz, and Rolando F Del Maestro. Artificial Intelligence Distinguishes Surgical Training Levels in a Virtual Reality Spinal Task. *The Journal of Bone and Joint Surgery: American volume*, 101(23):e127, 2019.

[3] Deo Rahul C. Machine Learning in Medicine. *Circulation*, 132(20):1920–1930, Nov 2015.

[4] Joseph A. Cruz and David S. Wishart. Applications of Machine Learning in Cancer Prediction and Prognosis. *Cancer Informatics*, 2:117693510600200030, Jan 2006.

[5] Ariel Kate Dubin, Danielle Julian, Alyssa Tanaka, Patricia Mattingly, and Roger Smith. A Model for Predicting the GEARS Score from Virtual Reality Surgical Simulator Metrics. *Surgical Endoscopy*, 32(8):3576–3581, 2018.

[6] Mohamed El Beheiry, Sébastien Doutreligne, Clément Caporal, Cécilia Ostertag, Maxime Dahan, and Jean-Baptiste Masson. Virtual Reality: Beyond Visualization. *Journal of Molecular Biology*, 431(7):1315–1321, 2019.

[7] Issam El Naqa and Martin J. Murphy. *What Is Machine Learning?*, page 3–11. Springer International Publishing, 2015.

[8] Mitchell S. Fourman, Hamid Ghaednia, Amanda Lans, Sophie Lloyd, Allison Sweeney, Kelsey Detels, Hidde Dijkstra, Jacobien H.F. Oosterhoff, Duncan C. Ramsey, Synho Do, and Joseph H. Schwab. Applications of Augmented and Virtual Reality in Spine Surgery and Education: A Review. *Seminars in Spine Surgery*, page 100875, 2021.

[9] Fabio Galbusera, Gloria Casaroli, and Tito Bassani. Artificial Intelligence and Machine Learning in Spine Research. *JOR SPINE*, 2(1):e1044, 2019.

[10] L. Humbert, J.A. De Guise, B. Aubert, B. Godbout, S. Parent, D. Mitton, and W. Skalli. 3D Reconstruction of the Spine From Biplanar X-Rays Using Longitudinal and Transversal Inferences. *Journal of Biomechanics*, 40:S160, 2007. Program and Abstracts of the XXI Congress, International Society of Biomechanics.

[11] C. Hung, W. Chen, P. Lai, C. Lin, and C. Lee. Comparing Deep Neural Network and Other Machine Learning Algorithms For Stroke Prediction in a Large-Scale Population-Based Electronic Medical Claims Database. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3110-3113, Jul 2017.

[12] WH Wan Ishak, Ku-Ruhana Ku-Mahamud, and Norita Md Norwawi. Mining Temporal Reservoir Data Using Sliding Window Technique. *CiiT International Journal of Data Mining Knowledge Engineering*, 3(8):473–478, 2011.

[13] Hua-Fu Li and Suh-Yin Lee. Mining Frequent Itemsets Over Data Streams Using Efficient Window Sliding Techniques. *Expert Systems With Applications*, 36(2):1466–1477, 2009.

[14] Yixuan Li and Zixuan Chen. Performance Evaluation of Machine Learning Methods for Breast Cancer Prediction. *Applied and Computational Mathematics*, 7(4):212, 2018.

[15] Ziad Obermeyer and Ezekiel J. Emanuel. Predicting the Future — Big Data, Machine Learning, and Clinical Medicine. *The New England Journal of Medicine*, 375(13):1216–1219, Sep 2016.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] Michael P Rogers, Anthony J DeSantis, Haroon Janjua, Tara M Barry, and Paul C Kuo. The Future Surgical Training Paradigm: Virtual Reality and Machine Learning in Surgical Education. *Surgery*, 2020.

[18] Pablo Ruiz. ML Approaches for Time Series. `https://towardsdatascience.com/ml-approaches-for\-time-series-4d44722e48fe`. 2019.

[19] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. Anomaly Detection in Streams With Extreme Value Theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1067–1075, 2017.

[20] Samuel B. Tomlinson, Benjamin K. Hendricks, and Aaron Cohen-Gadol. Immersive Three-Dimensional Modeling and Virtual Reality for Enhanced Visualization of Operative Neurosurgical Anatomy. *World Neurosurgery*, 131:313–320, 2019.

[21] Alfredo Vellido. The Importance of Interpretability and Visualization in Machine Learning for Applications in Medicine and Health Care. *Neural Computing and Applications*, pages 1–15, 2019.

[22] Edoardo Vignotto and Sebastian Engelke. Extreme Value Theory for Anomaly Detection–The GPD Classifier. *Extremes*, 23(4):501–520, 2020.

[23] Voinea, Gheorghe-Daniel and Butnariu, Silviu and Mogan, Gheorghe. Measurement and Geometric Modelling of Human Spine Posture for Medical Rehabilitation Purposes Using a Wearable Monitoring System Based on Inertial Sensors. *Sensors*, 17(1):1–19, 2017.

[24] Luqing Wang, Qinglin Zhao, Si Gao, Wei Zhang, and Li Feng. A New Extreme Detection Method for Remote Compound Extremes in Southeast China. *Frontiers in Earth Science*, 9:243, 2021.

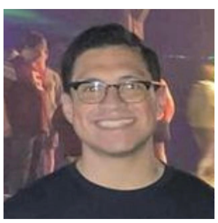
[25] Zach. How to Calculate Autocorrelation in Python. `https://www.statology.org/autocorrelation-python`. 2020.

Lin Hall is a candidate for the Master of Science in Data Science at East Carolina University, which he is on track to complete in May 2022. His main research interests involve machine learning and artificial intelligence and their utilization in AR/VR visualization. He also holds interests in automated systems and robotics, as well as atmospheric science. His research on this has resulted in one journal publication, but has also published two journals in the field of atmospheric science. He received finalist for Best Paper for the 2021 SEDE Conference, as well winning Best Master's Paper from the Southeastern Division of the Association of American Geographers in 2012. His research in atmospheric science also earned him the NASA Space Grant.

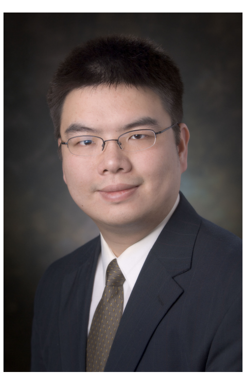
Ping Wang graduated in December 2021 with a Master of Science in Computer Science from ECU. Her research interest is machine learning. She currently works as a Data Analyst in Pennsylvania.

Grayson Blankenship graduated in May 2021 from East Carolina University with a Bachelor's in Computer Science. His research interests were with augmented and virtual reality, machine learning, and video games. He was supervised by Dr. Rui Wu. Grayson currently works as a QA Game Tester for Epic Games, and has held this position since Summer 2021.

Emmanuel Zenil Lopez graduated in May 2020 from East Carolina University with a Bachelor's in Computer Science. His research interests were with augmented and virtual reality, machine learning, and video games. He was supervised by Dr. Rui Wu.



Chris Castro graduated in May 2020 from East Carolina University with a Bachelor's in Computer Science. His research interests were with augmented and virtual reality, machine learning, and video games. He was supervised by Dr. Rui Wu.



Zhen Zhu is an associate professor at the Department of Engineering. Before joining ECU, he was a senior research engineer and a principal investigator with the Navigation Systems Division and the Advanced Concepts and Technologies Division in Northrop Grumman Electronic Systems from 2010 to 2013. From 2006 to 2010 he worked for the Ohio University Avionics Engineering Center as a senior research engineer, and he was also an adjunct assistant professor with the School of Electrical Engineering and Computer Science in Ohio University. He has developed various types of systems and algorithms for autonomous navigation and guidance of manned and unmanned aircraft; artificial intelligence and software radio.

Rui Wu received a Bachelor's degree in Computer Science and Technology from Jilin University, China in 2013. He then went on and received his Master and Ph.D. degrees in Computer Science and Engineering from the University of Nevada, Reno in 2015 and 2018, respectively. Rui is now working as an assistant professor in the Department of Computer Science at East Carolina University and collaborates with geological and hydrological scientists to protect the ecological system. His main research interests are machine learning and data visualization using AR/VR devices.