Applications of Bayesian Additive Regression for Cybersecurity

Rakesh Ravi K U (rk9cx)

Abstract

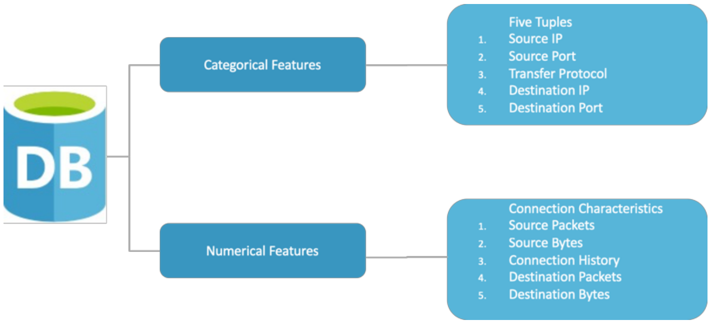
With the rise of internet usage, the advent of internet of things and the amount of data being transferred on a daily basis, our lives are becoming increasingly digitized. Most of our personal data is held inside digital devices and their connection to the internet makes them vulnerable to malicious attacks. As a result, Cybersecurity is one of the prime focus areas from a security standpoint for all developed and developing nations in the world. It is estimated that over $6 Billion Dollars will be spent on arming the United States with cybersecurity ammunition to counter malicious actors by 2021 [1]. Machine learning is being applied in cybersecurity to build robust systems in order to prevent hackers from attacking institutions and organizations. Several government institutions and research institutions are releasing data on malicious attacks that machine learning practitioners can use to build novel intrusion detection systems. Tree ensembles like gradient boosting and random forest have been used with a lot of success due to the ability of tree ensembles to handle high cardinality categorical variables efficiently. The only drawback of using such models is that the final prediction is a point estimate which prevents the practitioner from fully understanding the response variable. Bayesian additive regression trees (BART) has emerged as an alternative to gradient boosting and in many cases, delivers much better performs while providing distribution of probabilities instead of point estimates [2]. In this paper, I have implemented BART and compared its performance on the CTU-13 Stratosphere Project data set with other classification models.

Introduction

Out of all the traditional machine learning algorithms, sequential ensembles such as gradient boosting has worked really well with cybersecurity data sets. Despite delivering good performance, they have a few limitations that are hard to combat. Firstly, sequential ensembles are prone to overfitting on the training or the validation set. Secondly, there aren't too many options when it comes to regularization apart from the default hyperparameters in the model. Thirdly, like all other tree ensembles, sequential ensemble models are hard to interpret. Finally, the model provides a point estimate of the final prediction as opposed to a final distribution. BART is similar to Gradient Boosting Tree methods in that they sum the contribution of sequential weak learners. This is an alternative to the approach that random forest utilizes where there are independent learners and their responses are averaged. Gradient boosting uses a learning rate parameter to multiply each sequential tree and BART uses a prior distribution to append each sequential tree. Due to the Bayesian nature of the model, BART provides a distribution of the final prediction.

Data Description

I used an open source cybersecurity data set procured from Czech Technological University from a project named “stratosphere project” [3]. The data set contains simulated attacks as well as traffic that is benign and the data set was split into 70% train and 30% test/ validation set. The features in the data set include categorical variables such as IP addresses and ports to characterize the connections as well as numerical variables such as packets, bytes and



duration of the connection. The response variable is a binary class label that indicates whether a certain observation in the network traffic data is malicious or not. Figure above describes all the features in the data set.

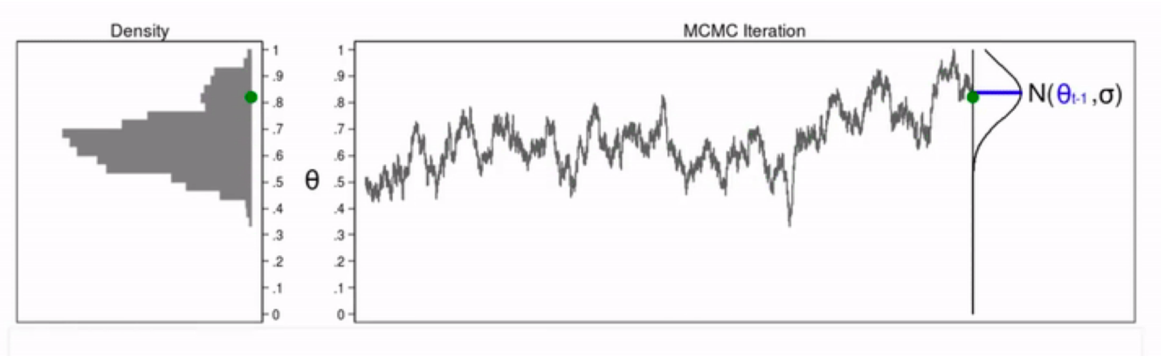
Methodology

BART is a sequential ensemble model that combines multiple weak learners that are built sequentially on residuals. They have four major features that help them deliver better performance than traditional machine learning models.

1. **Prior Assumptions** - The distribution of all the parameters are assumed to be in a certain distribution instead of using random values. Although prior distribution assumption is not very scientific, they reduce the time to train when the prior assumption is close to the actual distributions of the parameters. Through research, there are distributions that work better for each parameter of the sequential trees. In this model, there are three priors that are assumed before the modeling process.

    1. Tree Structure, T_i
    2. Leaf Parameters, M_i
    3. Error Variance, σ

2. **Markov Chain Monte Carlo (MCMC)** – As the model is sequential and each tree is dependent on the previous one. MCMC is a type of sampling where each subsequent sample draws information from the previous one and is a good fit for sequential trees. The figure below displays a MCMC simulation for a BART model where the proposed distribution is a normal distribution where the parametric mean depends on the mean of the previous iterations. This way, the sampling generally tends to converge to an equilibrium mean after which the parameters do not change too much.

    MCMC generally takes a long time to converge on its own because of which BART employs a convergence criterion.

3. **Metropolis Hastings Convergence Algorithm** - This algorithm works by generating a sequence of sample values in such a way that, as more and more samples are produced, the distribution of the sample values approximate the desired posterior distribution as accurately as possible. These samples are produced in an iterative fashion and there is a condition of dependence in each sample which causes a regularizing effect. Specifically, at each iteration, the algorithm picks the parameters for the next sample based on the current sample value. Then, based on a condition, the parameters for the next sample is either accepted or rejected. The condition for acceptance is based on the comparison between parameters for the next sample and parameters of the desired posterior distribution. In the first few hundred iterations, MCMC iterations generally waver a lot from the desired distributions and these are called burn-in iterations. Its recommended to not use the values obtained during the burn-in period.

4. **Bayesian Back fitting Algorithm** - The back fitting algorithm [4] is similar to back propagation used in deep learning and the following are the two major steps.

1. Given observed data (y), Bayesian setup induces a posterior using m successive draws of (T_j , M_j) conditionally on ($T_{(j)}$, $M_{(j)}$, σ).

2. Instead of estimating the joint probability distribution, the algorithm employs a Gibbs Sampling like method to estimate the join probability distribution from the conditional distribution.

1. $R_j = y - \sum_{k \neq j} g(x; T_k , M_k)$

2. Final draws are made successively out of (T_j / R_j , σ) and (M_j / T_j , R_j, σ)

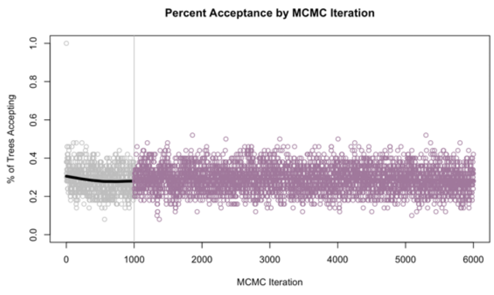
Results

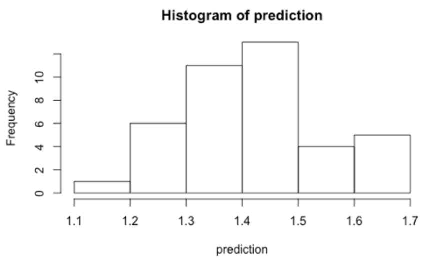
To prove that BART's efficiency and superior performance, I constructed an experiment to compare its performance with Gradient Boosting Classifier (GB) and Logistic Regression (LR). All the models were evaluated Accuracy, False Positive Rate, Precision and Recall.

The table below shows results of all models. BART outperforms GB and LR in all areas showing the lowest false positive rate which is a major concern in cybersecurity as a lot of resources gets wasted.

Metrics	Logistic Regression	Gradient Boosting Classifier	Bayesian Additive Regression Trees
Accuracy	**0.991**	**0.995**	**0.996**
False Positive Rate	**0.253**	**0.162**	**0.114**
Precision	**0.982**	**0.990**	**0.997**
Recall	**0.953**	**0.982**	**0.996**

The final BART model contained 6000 iterations with a burn-in iteration of 1000 that delivered the highest performance. Below is a figure(left) that shows the configuration used for the final model.





As explained earlier, BART provides a distribution for the prediction as show in the figure above on the right).

Conclusions

Through this project, I was able to prove that BART is a model that can perform better than most traditional machine learning models. Despite exhibiting good performance, I found a few limitations that could prevent the model from being used widely. Firstly, the model is computationally expensive to run and very difficult to scale to large data sets. Secondly, there is a huge dependence on the starting values or priors which are not easy to overcome. Thirdly, the implementation of BART is not available in programming languages other than R. In future work, I would work on implementing a parallelizable version of BART to extend this to more applications.

References

[1] "Cyberattacks are the fastest growing crime and predicted to cost the world $6 trillion annually by 2021." 2018. www.prnewswire.com/ne ws- releases/cyberattacks-are-the-fastest-growing-crime-and-predicte d-to-cost-the-world-6-trillion-annually-by-2021-300765090.html0765090.html. Accessed: March 5, 2019.

[2] Bayesian Additive Regression Trees "https://arxiv.org/pdf/0806.3286.pdf

[3] CTU -13 https://www.stratosphereips.org/datasets-ctu13

[4] Bayesian Backfittting "https://web.stanford.edu/~hastie/TALKS/gibbsgam.pdf"