Accelerating Hyperparameter Tuning in Machine Learning for Alzheimer's Disease With High Performance Computing

Fan Zhang[1,2]*, *Melissa Petersen*[1,2], *Leigh Johnson*[1,3], *James Hall*[1,3] and *Sid E. O'Bryant*[1,3]*

[1] *Institute for Translational Research, University of North Texas Health Science Center, Fort Worth, TX, United States,* [2] *Department of Family Medicine, University of North Texas Health Science Center, Fort Worth, TX, United States,* [3] *Department of Pharmacology and Neuroscience, University of North Texas Health Science Center, Fort Worth, TX, United States*

Edited by:
Reza Lashgari,
Shahid Beheshti University, Iran

Reviewed by:
Saman Sarraf,
Institute of Electrical and Electronics Engineers, United States
Xiaojie Zhao,
Rosalind Franklin University of Medicine and Science, United States
Daiwei Wan,
The First Affiliated Hospital of Soochow University, China

***Correspondence:**
Fan Zhang
fan.zhang@unthsc.edu
Sid E. O'Bryant
sid.obryant@unthsc.edu

Specialty section:
This article was submitted to Machine Learning and Artificial Intelligence, a section of the journal Frontiers in Artificial Intelligence

Received: *20 October 2021*
Accepted: *15 November 2021*
Published: *08 December 2021*

Citation:
Zhang F, Petersen M, Johnson L, Hall J and O'Bryant SE (2021) Accelerating Hyperparameter Tuning in Machine Learning for Alzheimer's Disease With High Performance Computing. Front. Artif. Intell. 4:798962. doi: 10.3389/frai.2021.798962

Driven by massive datasets that comprise biomarkers from both blood and magnetic resonance imaging (MRI), the need for advanced learning algorithms and accelerator architectures, such as GPUs and FPGAs has increased. Machine learning (ML) methods have delivered remarkable prediction for the early diagnosis of Alzheimer's disease (AD). Although ML has improved accuracy of AD prediction, the requirement for the complexity of algorithms in ML increases, for example, hyperparameters tuning, which in turn, increases its computational complexity. Thus, accelerating high performance ML for AD is an important research challenge facing these fields. This work reports a multicore high performance support vector machine (SVM) hyperparameter tuning workflow with 100 times repeated 5-fold cross-validation for speeding up ML for AD. For demonstration and evaluation purposes, the high performance hyperparameter tuning model was applied to public MRI data for AD and included demographic factors such as age, sex and education. Results showed that computational efficiency increased by 96%, which helped to shed light on future diagnostic AD biomarker applications. The high performance hyperparameter tuning model can also be applied to other ML algorithms such as random forest, logistic regression, xgboost, etc.

Keywords: machine learning, hyperparameter tuning, alzheimer's disease, high performance computing, support vector machine

INTRODUCTION

Alzheimer's disease (AD) is the most common form of dementia. In 2020, as many as 5.8 million Americans were living with AD. This number is projected to nearly triple by 2060 (Prevention, 2021). Machine Learning (ML) methods for AD and AD Related Dementias (ADRDs) is growing faster than ever before (Waring et al., 2008; Magnin et al., 2009; O'Bryant et al., 2011a; O'Bryant et al., 2011b; O'Bryant et al., 2013; O'Bryant et al., 2014; Weiner et al., 2015; O'Bryant et al., 2016; O'Bryant et al., 2017; Grassi et al., 2018; Hampel et al., 2018; O'Bryant et al., 2018; Stamate et al., 2019; Zetterberg and Burnham, 2019; Zhang and Sejdić, 2019; Franzmeier et al., 2020; O'Bryant et al., 2020; Rodriguez et al., 2021). A PubMed search using keywords of AD and ML showed that the number of publications related to ML for AD has increased by 146 percent from just two in 2006 to 294 in 2020. For example, O'Bryant et al. developed a Support Vector Machine (SVM) model with 398 plasma samples obtained from adults with Down syndrome to predict incident mild cognitive impairment

TABLE 1 | Talon3 computer nodes.

Quanity	Memory (GB)	Cores	Description
192	64	28	Dell PowerEdge C6320 server with two 2.4 GHz Intel Xeon E5-2680 v4 14-core processors
75	32	16	Dell PowerEdge R420 server with two 2.1 GHz Intel Xeon E5-2450 eight-core processors
64	64	16	Dell PowerEdge R420 server with two 2.1 GHz Intel Xeon E5-2450 eight-core processors
8	512	32	Dell PowerEdge R720 server with four 2.4 GHz Intel Xeon E5-4640 eight-core processors
16	64	28	Dell PowerEdge R730 server with two 2.4 GHz Intel Xeon E5-2680 v4 14-core processors and two Nvidia Tesla K80 GPUS (4,992 GPU cores/card)

TABLE 2 | Performance for testing set after hyperparameter tuning.

	Actual demented	Actual nondemented
Predicted demented	9	1
Predicted nondemented	3	16
Precision/PPV	90.00%	
Accuracy	86.21%	
Sensitivity	75.00%	
Specificity	94.12%	
NPV	84.21%	
AUC	90.80%	
PPV12	63.49%	
NPV12	96.50%	

(MCI) (AUC = 0.92) and incident AD (AUC = 0.88) (O'Bryant et al., 2020). O'Bryant et al. also developed a precision medicine model for targeted NSAID therapy in AD based on data collected from a previously conducted clinical trial. This work included 351 patients with mild-to-moderate AD that were enrolled into one of three trial arms: 1-year exposure to rofecoxib (25 mg once daily), naproxen (220 mg twice-daily) and placebo. The SVM model yielded 98% theragnostic accuracy in the rofecoxib arm and 97% accuracy in the naproxen arm, respectively (O'Bryant et al., 2018). Magnin et al. also built a SVM model with three-dimensional T1-weighted MR images of 16 patients with AD and 22 elderly controls and obtained a 94.5% mean accuracy for AD with a mean specificity of 96.6% and mean sensitivity of 91.5% (Magnin et al., 2009).

Improving speed and capability is a huge issue in applying ML to AD. It is possible that certain ML computations can be delayed because of the large amount of time to iteration that is required, for example, time to train with hyperparameters tuning. High performance computing (HPC) can be used to help meet the increasing demands for the speed and capabilities of processing ML for AD (Eddelbuettel, 2021). With the fast processing ability of high-performance computing systems, faster results can be delivered, which in turn would not only speed up finding the optimal hyperparameters for AD with ML models but would also identify opportunities to fix issues in hyperparameter tuning for AD ML models.

In this paper, based on the multicores parallel structure of Talon3 high performance computing provided by the University of North Texas, we present a high performance computing workflow to support our parallel SVM hyperparameter tuning. We applied the multicore high performance SVM hyperparameter tuning to 100 times repeated 5-fold cross-validation model for longitudinal MRI data of 150 subjects with 64 subjects classified as demented and 86 subjects classified as nondemented. The computational time was dramatically reduced by up to 96% for the high performance SVM hyperparameter tuning model. The multicores parallel structure and the high performance SVM hyperparameter tuning model can be used for other ML applications.

MATERIALS AND METHODS

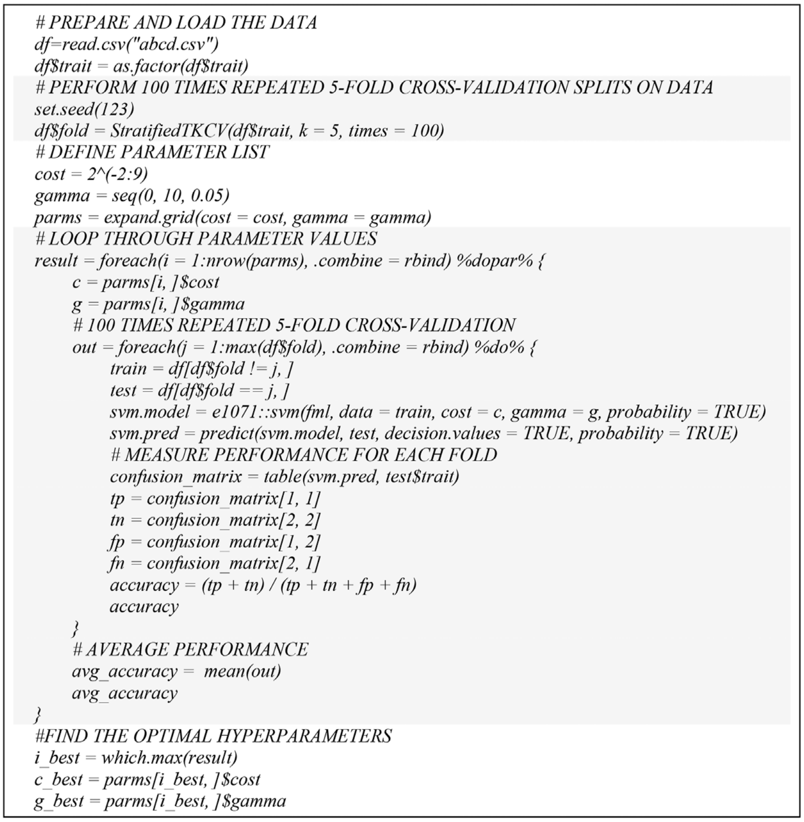
Parallel Structure

We used the Talon3 system (**Table 1**) provided by University of North Texas for this study due to its convenient computing services, which allowed us to import/export/execute large and complex parallel ML. The hardware configuration of the Talon3 contains the following: more than 8,300 CPU cores, 150,000 GPU cores, Mellanox FDR InfiniBand network, and over 1.4 Petabytes of Lustre File Storage. The amount of AD data necessary for performing ML with a PC workstation is massive. For example, in one study with 300 samples, to process just the 100 times repeated 5-fold cross-validation for hyperparameters tuning with SVM, it would require about 3 h of consecutive CPU time and 12 GB of storage with a local computer.

For parallel computing, the Talon3 provides several options including: SNOW, Rmpi, and multicore. We chose multicore because it executes parallel tasks on a single node as opposed to multiple nodes and the level of flexibility is higher than the other two options. For multicore parallel programming, submitting high performance ML includes two parts: a shell script and an R script. The shell script we submitted for multicore is for a single node with 28 cores in C6320. And for the R script high performance ML, we used doParallel (Michelle Wallig et al., 2020; Eddelbuettel, 2021) and foreach (Michelle Wallig and Steve, 2020; Eddelbuettel, 2021) packages.

Parallel SVM Hyperparameter Tuning

Based on the above parallel structure in Talon3 and doParallel package, we developed a high performance computing workflow to support our parallel SVM hyperparameter tuning (**Figure 1**). We used a grid search approach to find the best model parameters in terms of accuracy. This procedure mainly contains three steps: 1) define a grid to vary cost and gamma, 2) perform 100 times



```
# PREPARE AND LOAD THE DATA
df=read.csv("abcd.csv")
df$trait = as.factor(df$trait)
# PERFORM 100 TIMES REPEATED 5-FOLD CROSS-VALIDATION SPLITS ON DATA
set.seed(123)
df$fold = StratifiedTKCV(df$trait, k = 5, times = 100)
# DEFINE PARAMETER LIST
cost = 2^(-2:9)
gamma = seq(0, 10, 0.05)
parms = expand.grid(cost = cost, gamma = gamma)
# LOOP THROUGH PARAMETER VALUES
result = foreach(i = 1:nrow(parms), .combine = rbind) %dopar% {
    c = parms[i, ]$cost
    g = parms[i, ]$gamma
    # 100 TIMES REPEATED 5-FOLD CROSS-VALIDATION
    out = foreach(j = 1:max(df$fold), .combine = rbind) %do% {
        train = df[df$fold != j, ]
        test = df[df$fold == j, ]
        svm.model = e1071::svm(fml, data = train, cost = c, gamma = g, probability = TRUE)
        svm.pred = predict(svm.model, test, decision.values = TRUE, probability = TRUE)
        # MEASURE PERFORMANCE FOR EACH FOLD
        confusion_matrix = table(svm.pred, test$trait)
        tp = confusion_matrix[1, 1]
        tn = confusion_matrix[2, 2]
        fp = confusion_matrix[1, 2]
        fn = confusion_matrix[2, 1]
        accuracy = (tp + tn) / (tp + tn + fp + fn)
        accuracy
    }
    # AVERAGE PERFORMANCE
    avg_accuracy =  mean(out)
    avg_accuracy
}
#FIND THE OPTIMAL HYPERPARAMETERS
i_best = which.max(result)
c_best = parms[i_best, ]$cost
g_best = parms[i_best, ]$gamma
```

FIGURE 1 | Pseudo code for parallel SVM hyperparameter tuning.

repeated 5-fold cross-validation splits on training data, and 3) tune the cost and gamma of the SVM model.

100 Times Repeated 5-Fold Cross-Validation

A single run of the 5-fold cross-validation (O'Bryant et al., 2019) may result in a noisy estimate of model parameters. We adopted 100 times repeated 5-fold cross-validation (Kublanov et al., 2017) to improve the estimation of optimal parameters of the ML model. This involves simply repeating the cross-validation procedure 100 times and reporting the mean performance across all folds from all runs. This mean performance is then used for the determination of optimal parameters.

Metrics

The following eight measurements were involved in our evaluation: 1) Sensitivity (also called recall), the proportion of actual positive pairs that are correctly identified; 2) Specificity, the proportion of negative pairs that are correctly identified; 3) Precision, the probability of correct positive prediction; 4) Accuracy, the proportion of correctly predicted pairs; 5) Area Under the Curve; 6) Negative Predictive Value (NPV), the probability that subjects with a negative screening test truly don't have the disease; 7) Negative Predictive Value at base rate of 12% (NPV12); and 8) Positive Predictive Value at base rate of 12% (PPV12).

RESULTS

We downloaded open access longitudinal MRI data available on nondemented and demented older adults (Marcus et al., 2010a). The dataset consisted of longitudinal MRI data from 150 subjects aged 60 to 96. 72 of the subjects were classified as "nondemented" throughout the study. 64 of the subjects were classified as "demented" at the initial visit and remained so throughout the study. 14 subjects were classified as "nondemented" at the initial visit and were subsequently characterized as "demented"at a later study visit. For each subject, three to four individual T1-weighted magnetization prepared rapid gradient-echo (MP-RAGE) images were acquired in a single imaging session (Marcus et al., 2010b). The subject-independent model we developed for parallel hyperparameter tuning is not based on a classifier trained for each subject individually. We chose the following five imaging and clinical variables to predict the status of AD: SES (Socioeconomic Status), MMSE (Mini Mental State Examination), eTIV (Estimated Total Intracranial Volume), nWBV (Normalize Whole Brain Volume), and ASF (Atlas Scaling Factor). Measurements of these variables in this cohort

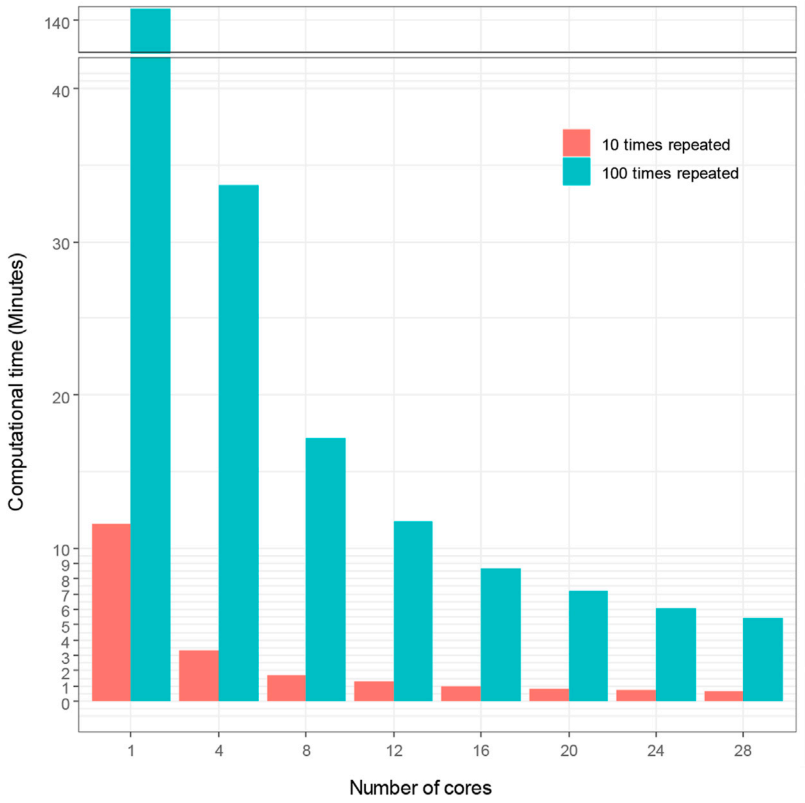


FIGURE 2 | Computational time vs. number of cores with SVM modeling.

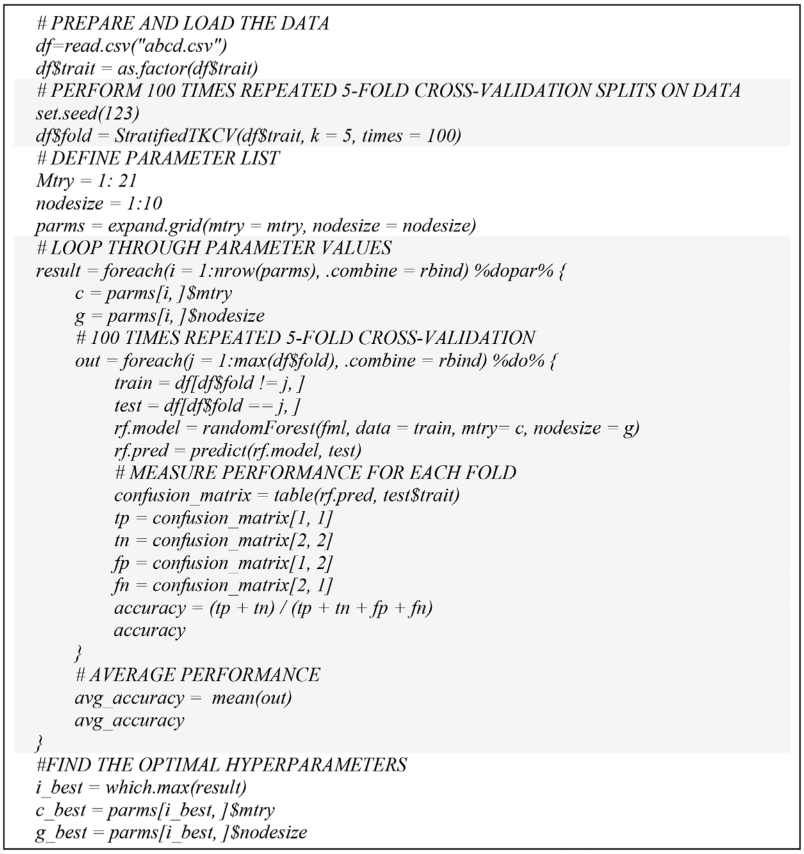
including clinical dementia rating scale (CDR), nWBV, eTIV, ASF, etc. have been previously described elsewhere (Marcus et al., 2010b). Three demographic variables: Sex, Age, and Edu (Years of education) were also added as covariates.

We used sbatch commands to submit shell scripts and R scripts for comparing computational time for hyperparameter tuning under different number of cores and repeated times of 5-fold cross-validation in Talon3. With SVM modeling, we demonstrate in **Figure 2** how the number of cores affects the computational time of hyperparameter tuning, which shows that the computational time decreases proportionally as the number of cores increases. **Figure 2** also demonstrates that the repeated times of the 5-fold cross-validation algorithm for hyperparameter affects the computational time. The computational time increased proportionally with the increasing repeated times. The time spent initially for the hyperparameter tuning without using high performance computing is very large (140.73 min for t = 100; 11.58 min for t = 10). The computational time decreased to 5.44 and 0.67 min, for t = 100 and for t = 10 respectively, when we used 28 cores to accelerate hyperparameter tuning in ML. We thereby reduced computational time by up to 96%, with high performance ML model.

The optimal hyperparameters we obtained are the same for all runs (gamma = 0.005, cost = 32). We used grid search method and set boundary for the two parameters: cost and gamma as suggested in the paper (Hsu et al., 2003) where fine grid search was on cost = (2, 32) and gamma = [2(−7), 2(−3)]. We extended the cost boundary to (0.25, 512) and the gamma boundary to (0, 10) to catch as much change as possible. Variables importance under the SVM model with the optimal hyperparameters shows that the MMSE, nWBV, and SES are leading variables in predicting dementia (AD) status. Out of the three demographic variables, education was shown to be less important for the SVM model than Age and Sex.

With the optimal hyperparameters, the average performance that the SVM model achieved for a testing set of 12 Demented and 17 Nondemented is reported on below for both 100 times repeated 5 -fold cross-validation and 10 times repeated 5 -fold cross-validation. The performance (**Table 2**) is slightly higher than previously reported at https://www.kaggle.com/hyunseokc/detecting-early-alzheimer-s, which achieved accuracy = 0.82, sensitivity = 0.70, and AUC = 0.82 for SVM. Our results show that the high performance SVM hyperparameter tuning workflow that we presented can significantly reduce computational time while maintaining the necessary accuracy.

In order to demonstrate the extensibility of our hyperparameter tuning workflow to other ML models, we also followed the SVM hyperparameter tuning workflow (**Figure 3**) and adopted random forest into our parallel hyperparameter tuning workflow (**Figure 4**). We obtained consistent results that the computation time for hyperparameter tuning of random forest was also remarkedly reduced (**Figure 5**). The computational time was reduced from 47.67 to 2.24 min by



```
# PREPARE AND LOAD THE DATA
df=read.csv("abcd.csv")
df$trait = as.factor(df$trait)
# PERFORM 100 TIMES REPEATED 5-FOLD CROSS-VALIDATION SPLITS ON DATA
set.seed(123)
df$fold = StratifiedTKCV(df$trait, k = 5, times = 100)
# DEFINE PARAMETER LIST
Mtry = 1: 21
nodesize = 1:10
parms = expand.grid(mtry = mtry, nodesize = nodesize)
# LOOP THROUGH PARAMETER VALUES
result = foreach(i = 1:nrow(parms), .combine = rbind) %dopar% {
      c = parms[i, ]$mtry
      g = parms[i, ]$nodesize
      # 100 TIMES REPEATED 5-FOLD CROSS-VALIDATION
      out = foreach(j = 1:max(df$fold), .combine = rbind) %do% {
            train = df[df$fold != j, ]
            test = df[df$fold == j, ]
            rf.model = randomForest(fml, data = train, mtry= c, nodesize = g)
            rf.pred = predict(rf.model, test)
            # MEASURE PERFORMANCE FOR EACH FOLD
            confusion_matrix = table(rf.pred, test$trait)
            tp = confusion_matrix[1, 1]
            tn = confusion_matrix[2, 2]
            fp = confusion_matrix[1, 2]
            fn = confusion_matrix[2, 1]
            accuracy = (tp + tn) / (tp + tn + fp + fn)
            accuracy
      }
      # AVERAGE PERFORMANCE
      avg_accuracy =  mean(out)
      avg_accuracy
}
#FIND THE OPTIMAL HYPERPARAMETERS
i_best = which.max(result)
c_best = parms[i_best, ]$mtry
g_best = parms[i_best, ]$nodesize
```

FIGURE 3 | Pseudo code for parallel RF hyperparameter tuning.

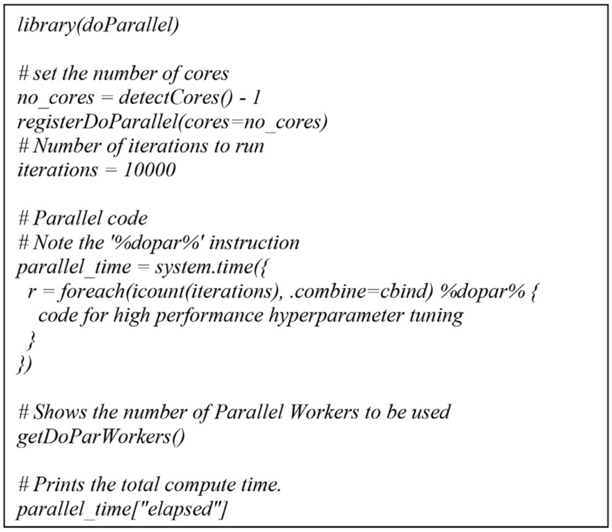
95% and from 5.25 min to 18.17 s by 94%, for *t* = 100 and *t* = 10 respectively.

We also tested the adaptability of our hyperparameter tuning workflow to the Texas Alzheimer's Research and Care Consortium (TARCC) dataset (Zhang et al., 2021). The TARCC dataset contains a total of 300 cases (150 AD cases; 150 Normal Control cases). Each subject (at one of the five participating TARCC sites) undergoes an annual standardized assessment, which includes a medical evaluation, neuropsychological testing, and a blood draw. The same blood-based biomarkers in (Zhang et al., 2021) were used as features for parallel hyperparameter tuning. Even when adopting a new TARCC dataset into our parallel hyperparameter tuning workflow (**Figure 1**), we obtained consistent results, which showed that the computation time for the hyperparameter tuning of the new TARCC dataset was also remarkedly reduced by about 96%. The computational time was reduced from 311.5 to 12.48 h by 96% and from 34.03 to 1.6 h by 95%, for *t* = 100 and *t* = 10 respectively.

DISCUSSIONS

HPC advances have successfully helped scientists and researchers to achieve various breakthrough innovations in the field of Omics-medicine, technology, retail, banking and so on (Merelli et al., 2014). For example, HPC has been applied to Next Generation Sequencing that is extremely data-intensive and needs ultra-powerful workstations to process the ever-growing data (Schmidt and Hildebrandt, 2017). Hyperparameter tuning component of ML can be a high-performance computing problem as it requires a large amount of computation and data motion. ML requires a computationally-intensive grid search and lots of computational power to help enable faster tuning cycles. Introducing HPC to ML can take advantage of high volumes of data as well as speed up the process of hyperparameter tuning.

Therefore, we presented a parallel hyperparameter tuning workflow with HPC to exploit modern parallel infrastructures to execute large-scale calculations by simultaneously using multiple compute resources. The rationales are 1) the foreach package that the workflow is based on supports parallel execution and provides a new looping construct for executing R code repeatedly. Specifically, a problem is broken into discrete parts that can be solved concurrently and an overall control/coordination mechanism is employed; 2) the foreach package can be used with a variety of different parallel computing systems, include NetWorkSpaces and snow; and 3) foreach can be used with iterators, which allows the data to be specified in a very flexible way.



```
library(doParallel)

# set the number of cores
no_cores = detectCores() - 1
registerDoParallel(cores=no_cores)
# Number of iterations to run
iterations = 10000

# Parallel code
# Note the '%dopar%' instruction
parallel_time = system.time({
  r = foreach(icount(iterations), .combine=cbind) %dopar% {
    code for high performance hyperparameter tuning
  }
})

# Shows the number of Parallel Workers to be used
getDoParWorkers()

# Prints the total compute time.
parallel_time["elapsed"]
```

FIGURE 4 | R script for parallel computing.

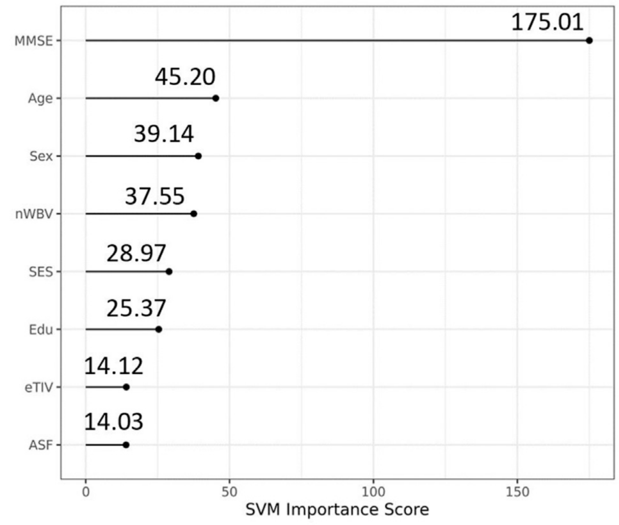
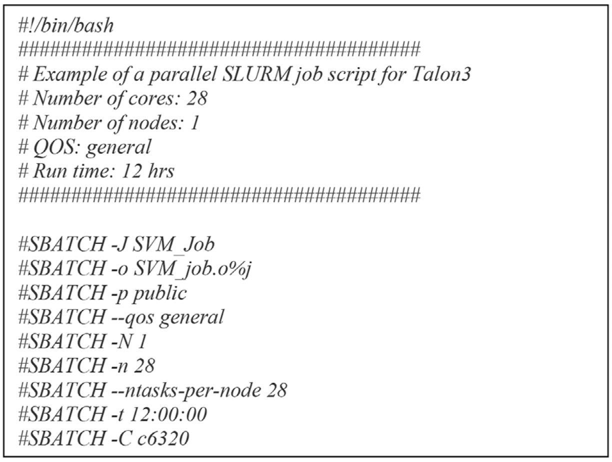


FIGURE 5 | Variable importance of the eight variables.

The multicore high performance SVM hyperparameter tuning workflow we presented is hardware-agnostic and can be used in HPCs of most U.S. universities or commerical clouds for example, Amazon AWS, Microsoft Azure, Google Cloud, etc. Before executing the multicores high performance SVM hyperparameter tuning, R package (V4.0.3, Linux) and doParallel and foreach libraries should be installed successfully, which are met for HPCs in most U.S. universities or commerical clouds. There are mainly two of the most popular job schedulers used for requesting resources allocations on a multi user cluster: 1) the Simple Linux Utility for Resource Management (Slurm) and 2) the Portable Batch System (PBS). In **Figure 6**, we described the shell script for parallel computing for the Slurm system. Similary a shell script for parallel computing for PBS system is as followed.



```
#!/bin/bash
##########################################
# Example of a parallel SLURM job script for Talon3
# Number of cores: 28
# Number of nodes: 1
# QOS: general
# Run time: 12 hrs
##########################################

#SBATCH -J SVM_Job
#SBATCH -o SVM_job.o%j
#SBATCH -p public
#SBATCH --qos general
#SBATCH -N 1
#SBATCH -n 28
#SBATCH --ntasks-per-node 28
#SBATCH -t 12:00:00
#SBATCH -C c6320
```

FIGURE 6 | Shell script for parallel computing.

The multicore high performance SVM hyperparameter tuning workflow significantly reduced computational time while maintaining a consistent detection accuracy. The workflow was diagrammed through a multicore computing pseudo code using the doParallel package in R for high performance hyperparameter tuning. The basic idea of multicore computing is to allow a single program, in this case R, to run multiple threads simultaneously in order to reduce the "walltime" required for completion. The doParallel package in R is one of several "parallel backends" for the foreach. It establishes communication between multiple cores, even on different physical "nodes" linked by network connections. The foreach function evaluates an expression for each value of the counter (iterator) "case". The %dopar% operator is used to execute the code in parallel. Using %do% instead would lead to sequential computation by the primary process. When parallelizing nesting for loops, there is always a question of which loop to parallelize. If the task and number of iterations vary in size, then it's really hard to know which loop to parallelize. We parallelized the outer loop in our SVM hyperparameter tuning because this would result in larger individual tasks, and larger tasks can often be performed more efficiently than smaller tasks. The hyperparameter tuning could be parallelized at the inner loop also if the outer loop doesn't have many iterations and the tasks are already large.

The multicores high performance hyperparameter tuning workflow can also be used for other ML such as random forest, logistic regression, xgboost, etc. For example, we demonstrated that a random forest model can be adopted into our parallel hyperparameter tuning model (**Figure 3**) and the results we obtained were consistent in that the computation time for hyperparameter tuning of random forest models were remarkedly reduced (**Figure 7**). In the future, we plan to use Rmpi library to create multinodes parallel computing workflow for hyperparameter tuning when Talon3 supports multinodes parallel computing to run R script.

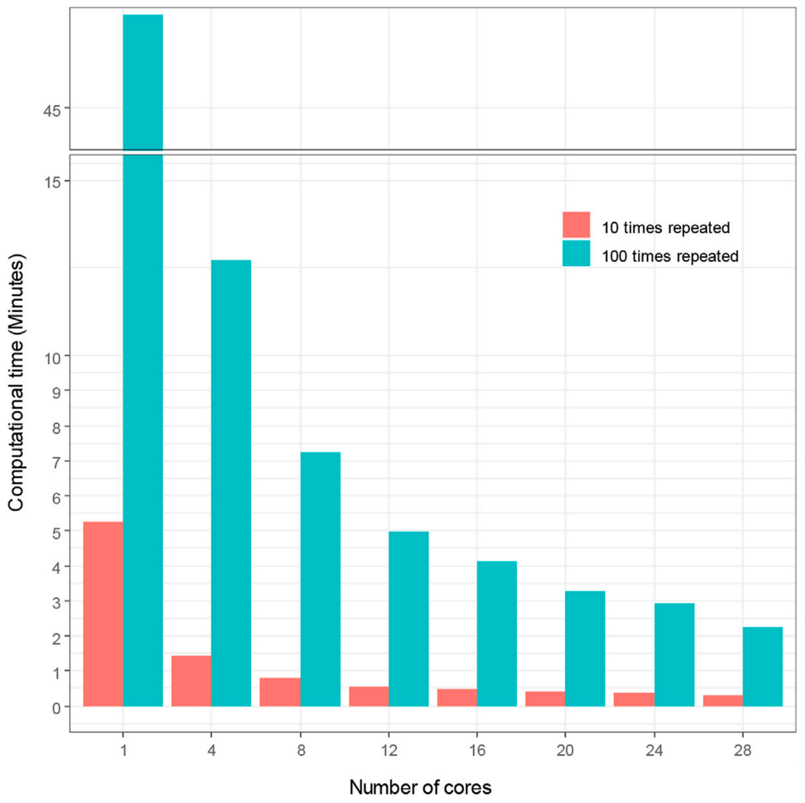


FIGURE 7 | Computational time vs. number of cores with RF modeling.

Our optimal hyperparameter model also showed that MMSE, Age, Sex, nWBV, and SES are important variables in AD diagnosis, which is consistent with previous findings. For example, Arevalo-Rodriguez et al. found that baseline MMSE scores can achieve a sensitivity of 76% and specificity of 94% for predicting conversion from MCI to dementia (in general) and a sensitivity of 89% and specificity of 90% for predicting conversion from MCI to AD dementia (Arevalo-Rodriguez et al., 2015). Advanced age and sex are two of the most prominent risk factors for dementia. Females are more likely to be susceptible for developing AD dementia than males (Podcasy and Epperson, 2016). Podcasy at Penn PROMOTES Research on Sex in Health and examined sex and gender differences in the development of dementia and suggested that researchers should consider sex as a biological variable for dementia research (Podcasy and Epperson, 2016). Rose et al. evaluated the combination of cerebrospinal fluid biomarkers with education and normalized whole-brain volume (nWBV) to predict incident cognitive impairment (Roe et al., 2011). They concluded that time to incident of cognitive impairment is moderated by education and nWBV for individuals with normal cognition had higher levels of cerebrospinal fluid tau and ptau at baseline (Roe et al., 2011). Khan et al. and Leong et al. assessed the role of various features on the prognosis of AD, and found that sex, age, MMSE, nWBV, and SES were significantly associated with and made an impact on the occurrence of AD (Leong and Abdullah, 2019; Khan and Zubair, 2020).

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://www.oasis-brains.org.

AUTHOR CONTRIBUTIONS

Conception and design of study: FZ and SO'B. Acquisition, and analysis of data: SO'B, FZ, and MP. Drafting manuscript or figures: SO'B, FZ, LJ, MP, and JH.

FUNDING

Research reported in this publication was supported by the National Institute on Aging of the National Institutes of Health under Award Number R01AG058537, R01AG054073, R01AG058533, and 3R01AG058533-02S1. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

ACKNOWLEDGMENTS

The authors acknowledge the North Texas Scientific Computing, a division under the office of the CIO at University of North Texas, for providing Talon computational and data analytics resources that have contributed to the research results reported within this paper. Data were provided by OASIS: Longitudinal: Principal Investigators: D. Marcus, R. Buckner, J. Csernansky, J. Morris; P50 AG05681, P01 AG03991, P01 AG026276, R01 AG021910, P20 MH071616, U24 RR021382.

REFERENCES

Arevalo-Rodriguez, I., Smailagic, N., Roqué I Figuls, M., Ciapponi, A., Sanchez-Perez, E., Giannakou, A., et al. (2015). Mini-Mental State Examination (MMSE) for the Detection of Alzheimer's Disease and Other Dementias in People with Mild Cognitive Impairment (MCI). *Cochrane Database Syst. Rev.* 2015, CD010783. doi:10.1002/14651858.CD010783.pub2

Eddelbuettel, D. (2021). *Parallel Computing R. A. Brief Review* 13, e1515. doi:10.1002/wics.1515

Franzmeier, N., Koutsouleris, N., Benzinger, T., Goate, A., Karch, C. M., Fagan, A. M., et al. (2020). Alzheimer's Disease Neuroimaging, I., Dominantly Inherited AlzheimerPredicting Sporadic Alzheimer's Disease Progression *via* Inherited Alzheimer's Disease-informed Machine-learning. *Alzheimer's Demen.* 16, 501–511. doi:10.1002/alz.12032

Grassi, M., Perna, G., Caldirola, D., Schruers, K., Duara, R., and Loewenstein, D. A. (2018). A Clinically-Translatable Machine Learning Algorithm for the Prediction of Alzheimer's Disease Conversion in Individuals with Mild and Premild Cognitive Impairment. *Jad* 61, 1555–1573. doi:10.3233/jad-170547

Hampel, H., O'Bryant, S. E., Molinuevo, J. L., Zetterberg, H., Masters, C. L., Lista, S., et al. (2018). Blood-based Biomarkers for Alzheimer Disease: Mapping the Road to the Clinic. *Nat. Rev. Neurol.* 14, 639–652. doi:10.1038/s41582-018-0079-7

Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2003). A Practical Guide to Support Vector Classification Chih-Wei Hsu. Chih-Chung Chang, and Chih-Jen Lin. Available at: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

Khan, A., and Zubair, S. (2020). Longitudinal Magnetic Resonance Imaging as a Potential Correlate in the Diagnosis of Alzheimer Disease: Exploratory Data Analysis. *JMIR Biomed. Eng.* 5, e14389. doi:10.2196/14389

Kublanov, V. S., Dolganov, A. Y., Belo, D., and Gamboa, H. (2017). Comparison of Machine Learning Methods for the Arterial Hypertension Diagnostics. *Appl. Bionics Biomech.* 2017, 5985479. doi:10.1155/2017/5985479

Leong, L. K., and Abdullah, A. A. (2019). Prediction of Alzheimer's Disease (AD) Using Machine Learning Techniques with Boruta Algorithm as Feature Selection Method. *J. Phys. Conf. Ser.* 1372, 012065. doi:10.1088/1742-6596/1372/1/012065

Magnin, B., Mesrob, L., Kinkingnéhun, S., Pélégrini-Issac, M., Colliot, O., Sarazin, M., et al. (2009). Support Vector Machine-Based Classification of Alzheimer's Disease from Whole-Brain Anatomical MRI. *Neuroradiology* 51, 73–83. doi:10.1007/s00234-008-0463-x

Marcus, D. S., Fotenos, A. F., Csernansky, J. G., Morris, J. C., and Buckner, R. L. (2010a). Open Access Series of Imaging Studies: Longitudinal MRI Data in Nondemented and Demented Older Adults. *J. Cogn. Neurosci.* 22, 2677–2684. doi:10.1162/jocn.2009.21407

Merelli, I., Pérez-Sánchez, H., Gesing, S., and D'agostino, D. (2014). High-performance Computing and Big Data in Omics-Based Medicine. *Biomed. Res. Int.* 2014, 825649. doi:10.1155/2014/825649

Michelle Wallig, M. C., Steve, W., and Dan, T. (2020). doParallel: Foreach Parallel Adaptor for the 'parallel' Package. R package version 1.0.16. Available at: https://cran.r-project.org/web/packages/doParallel/index.html.

Michelle Wallig, M., and Steve, W. (2020). foreach: Provides Foreach Looping Construct R package version 1.5.1. Available at: https://cran.r-project.org/web/packages/foreach/index.html.

O'bryant, S. E., Zhang, F., Johnson, L. A., Hall, J., Edwards, M., Grammas, P., et al. (2018). A Precision Medicine Model for Targeted NSAID Therapy in Alzheimer's Disease. *J. Alzheimers Dis.* 66, 97–104. doi:10.3233/JAD-180619

O'bryant, S. E., Zhang, F., Silverman, W., Lee, J. H., Krinsky-Mchale, S. J., Pang, D., et al. (2020). Proteomic Profiles of Incident Mild Cognitive Impairment and Alzheimer's Disease Among Adults with Down Syndrome. *Alzheimers Dement (Amst)* 12, e12033. doi:10.1002/dad2.12033

O'bryant, S. E., Edwards, M., Johnson, L., Hall, J., Villarreal, A. E., Britton, G. B., et al. (2016). A Blood Screening Test for Alzheimer's Disease. *Alzheimer's Demen. Diagn. Assess. Dis. Monit.* 3, 83–90. doi:10.1016/j.dadm.2016.06.004

O'bryant, S. E., Edwards, M., Zhang, F., Johnson, L. A., Hall, J., Kuras, Y., et al. (2019). Potential Two-step Proteomic Signature for Parkinson's Disease: Pilot Analysis in the Harvard Biomarkers Study. *Alzheimer's Demen. Diagn. Assess. Dis. Monit.* 11, 374–382. doi:10.1016/j.dadm.2019.03.001

O'bryant, S. E., Mielke, M. M., Rissman, R. A., Lista, S., Vanderstichele, H., Zetterberg, H., et al. (2017). Blood-based Biomarkers in Alzheimer Disease: Current State of the Science and a Novel Collaborative Paradigm for Advancing from Discovery to clinicBlood-Based Biomarkers in Alzheimer Disease: Current State of the Science and a Novel Collaborative Paradigm for Advancing from Discovery to Clinic. *Alzheimer's Demen.* 13, 45–58. doi:10.1016/j.jalz.2016.09.014

O'bryant, S. E., Xiao, G., Barber, R., Huebinger, R., Wilhelmsen, K., Edwards, M., et al. (2011a). Texas Alzheimer's, R., Care, C., and Alzheimer's Disease Neuroimaging, IA Blood-Based Screening Tool for Alzheimer's Disease that Spans Serum and Plasma: Findings from TARC and ADNI. *PLoS One* 6, e28092. doi:10.1371/journal.pone.0028092

O'bryant, S. E., Xiao, G., Barber, R., Reisch, J., Hall, J., Cullum, C. M., et al. (2011b). Texas Alzheimer'sA Blood-Based Algorithm for the Detection of Alzheimer's Disease. *Dement Geriatr. Cogn. Disord.* 32, 55–62. doi:10.1159/000330750

O'bryant, S. E., Xiao, G., Edwards, M., Devous, M., Gupta, V. B., Martins, R., et al. (2013). Texas Alzheimer's, R., and Care, CBiomarkers of Alzheimer's Disease Among Mexican Americans. *Jad* 34, 841–849. doi:10.3233/jad-122074

O'bryant, S. E., Xiao, G., Zhang, F., Edwards, M., German, D. C., Yin, X., et al. (2014). Validation of a Serum Screen for Alzheimer's Disease across Assay Platforms, Species, and Tissues. *Jad* 42, 1325–1335. doi:10.3233/jad-141041

Podcasy, J. L., and Epperson, C. N. (2016). Considering Sex and Gender in Alzheimer Disease and Other Dementias. *Dialogues Clin. Neurosci.* 18, 437–446. doi:10.31887/DCNS.2016.18.4/cepperson

Prevention, C. F. D. C. A. (2021). Alzheimer's Disease and Healthy Aging. Available at: https://www.cdc.gov/aging/aginginfo/alzheimers.html.

Rodriguez, S., Hug, C., Todorov, P., Moret, N., Boswell, S. A., Evans, K., et al. (2021). Machine Learning Identifies Candidates for Drug Repurposing in Alzheimer's Disease. *Nat. Commun.* 12, 1033. doi:10.1038/s41467-021-21330-0

Roe, C. M., Fagan, A. M., Grant, E. A., Marcus, D. S., Benzinger, T. L., Mintun, M. A., et al. (2011). Cerebrospinal Fluid Biomarkers, Education, Brain Volume, and Future Cognition. *Arch. Neurol.* 68, 1145–1151. doi:10.1001/archneurol.2011.192

Schmidt, B., and Hildebrandt, A. (2017). Next-generation Sequencing: Big Data Meets High Performance Computing. *Drug Discov. Today* 22, 712–717. doi:10.1016/j.drudis.2017.01.014

Stamate, D., Kim, M., Proitsi, P., Westwood, S., Baird, A., Nevado-Holgado, A., et al. (2019). A Metabolite-based Machine Learning Approach to Diagnose Alzheimer-type Dementia in Blood: Results from the European Medical Information Framework for Alzheimer Disease Biomarker Discovery Cohort. *Alzheimer's Demen. Translational Res. Clin. Interventions* 5, 933–938. doi:10.1016/j.trci.2019.11.001

Waring, S., O'bryant, S., Reisch, J., Diaz-Arrastia, R., Knebl, J., and Doody, R. (2008). The Texas Alzheimer's Research Consortium Longitudinal Research Cohort: Study Design and Baseline Characteristics. *Tex. Public Health J* 60, 9–13.

Weiner, M. W., Veitch, D. P., Aisen, P. S., Beckett, L. A., Cairns, N. J., Cedarbaum, J., et al. (2015). 2014 Update of the Alzheimer's Disease Neuroimaging

Initiative: A Review of Papers Published since its Inception. *Alzheimers Dement* 11, e1–120. doi:10.1016/j.jalz.2014.11.001

Zetterberg, H., and Burnham, S. C. (2019). Blood-based Molecular Biomarkers for Alzheimer's Disease. *Mol. Brain* 12, 26. doi:10.1186/s13041-019-0448-1

Zhang, F., Petersen, M., Johnson, L., Hall, J., and O'Bryant, S. E. (2021). Recursive Support Vector Machine Biomarker Selection for Alzheimer's Disease. *Jad* 79, 1691–1700. doi:10.3233/jad-201254

Zhang, Z., and Sejdić, E. (2019). Radiological Images and Machine Learning: Trends, Perspectives, and Prospects. *Comput. Biol. Med.* 108, 354–370. doi:10.1016/j.compbiomed.2019.02.017

Conflict of Interest: SO'B has multiple pending and issued patents on blood biomarkers for detecting and precision medicine therapeutics in neurodegenerative diseases. He is a founding scientist and owns stock options in Cx Precision Medicine, Inc.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.