AN INTEGRATIVE MACHINE LEARNING APPROACH FOR SMALL SAMPLES AND HIGH-DIMENSIONAL IMBALANCED DATA IN PSYCHOLOGICAL EXPERIMENT

Chaewon Lee

A thesis submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Arts in the Department of Psychology and Neuroscience (Quantitative Psychology)

Chapel Hill
2023

Approved by:

Kathleen Gates

Daniel Bauer

Kenneth Bollen

ABSTRACT

Chaewon Lee: An integrative machine learning approach for small samples and high-dimensional imbalanced data in psychological experiment
(Under the direction of Kathleen Gates)

Machine learning for classification may not be immediately useful for many contexts seen in psychology. Psychological data often limit its efficacy due to small sample size, high dimensionality, and class imbalance. The current study presents an integrative machine learning approach that can be a useful solution to the challenges encountered when the aforementioned issues are inherent in psychological data. The tested approach consists of three consecutive steps – feature selection, minority oversampling, and predictive modeling. To begin with, feature selection tackles high dimensionality and extracts important features out of original predictors, using elastic net logistic regression. Then, synthetic minority oversampling technique addresses class imbalance, generating new observations primarily for the minority class. Finally, supervised machine learning algorithms build predictive models, using the oversampled feature set. The algorithms employed in this study include support vector machine, extreme gradient boosting, deep neural network, and logistic regression. They fully exploit the small sample with leave-one-out cross-validation. The current study demonstrates the utility of the integrative classification approach with an empirical analysis on predicting suicide attempt by a sample of patients diagnosed with bipolar I disorder, using their event-related potentials (ERPs). The study shows how prediction can be improved by the integrative modeling as the first two analytical steps being added to the generic process of predictive modeling.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACC	Accuracy
AUC	Area under the curve
BD	Bipolar disorder
DNN	Deep neural network
EEG	Electroencephalography
ENR	Elastic net logistic regression
ERP	Event-related potentials
LOOCV	Leave-one-out cross-validation
LR	Logistic regression
ML	Machine learning
NSA	Suicide non-attempters
PRC	Precision
RF	Random Forest
ROC	Receiver operating characteristic
SA	Suicide attempters
SGD	Stochastic gradient descent
SMOTE	Synthetic minority oversampling technique
SNT	Sensitivity
SPC	Specificity
SVM	Support vector machine
XGBoost	Extreme gradient boosting

CHAPTER 1: INTRODUCTION

Obstacles to Using Machine Learning in Psychology

Machine learning (ML) approaches for classification may not be immediately useful in psychology. It is largely attributed to several unavoidable issues that are often inherent in psychological data such as small sample size, high dimensionality, and class imbalance. Above all, small samples are commonplace in psychological data since data acquisition from human participants or lab experiments tends to be cost-heavy, and most psychology research targets specific groups rather than the general population (Button et al., 2013; Muth et al., 2016; Vabalas et al., 2019). Such aspects in psychological data may circumvent the utility of ML algorithms out-of-the-box, as many of them demand large samples for effectual prediction (Jiang et al., 2020). In addition, analyses on small samples may pose a risk that prediction results are highly variable upon the entry of new data. Meanwhile, psychological data gets increasingly high-dimensional with the number of predictors being close to or exceeding the number of observations. It is mainly because smartphones and wearable technologies have recently emerged as the main collecting tools for a wide array of passive data gathered without direct involvement of participants (Maher et al., 2019). Due to these reasons, small samples with high dimensionality are very common in psychology.

In case of high dimensionality and small sample size, overfitting is a key challenge (Shen et al., 2022; Vabalas et al., 2019). Overfitting occurs when a ML algorithm generates a well-fitting

model on training data but a poorly generalizable model on test data (Yeom et al., 2018; Ying, 2019). Such a biased prediction gets more pronounced in the presence of class imbalance (Li et al., 2020). In binary classification, we mainly attempt to identify a target group of interest (e.g., patients with diseases), namely positive cases. But given the prevalence of negative cases (e.g., healthy control), ML algorithms would be partial to detecting the negative cases well, while poorly capturing the positive cases. In that negative cases often outnumbers positive cases in psychological data, exploring an effective way to address class imbalance would benefit ML algorithms to better identify the group of interest that researchers are ultimately trying to predict (Jacobucci & Li, 2022).

Goal of Study

The current study demonstrates the utility of an integrative ML approach for classification with an empirical analysis on predicting suicide attempt by a sample of participants diagnosed with bipolar I disorder, using their event-related potentials (ERPs). This research consists of three analyses. Analysis 1 explores the efficacy of feature selection that serves as the first step of the integrative approach in prediction. Analysis 2 implements the integrative classification approach that consists of feature selection, minority oversampling, and predictive modeling and explores how prediction can be further improved as minority oversampling process (SMOTE) is added to analysis 1 even when sample sizes for two classes are not seriously imbalanced. Analysis 3 looks into whether ERPs serve as superior predictors of suicide attempt to demographic variables. The present study is organized as follows. The remaining part of introduction discusses the severity of suicide attempt by BD patients and reviews previous ML literature of which research goal was to predict a person's suicide risk. Chapter 2 introduces the empirical data – N200 peak latency ERPs

and variables. Chapter 3 accounts for supervised ML algorithms that were put in use for feature selection and predictive modeling. It also explores two essential statistical methods paired with ML algorithms – leave-one-out cross-validation (LOOCV) and synthetic minority oversampling technique (SMOTE) – and gives the rationales for employing them in the present study. Also, the evaluation metrics that were used to assess the prediction performance of ML algorithms are introduced. Chapter 4 presents the overall prediction results across three analyses and shows how the integrative ML approach effectively improved prediction. Chapter 5 discusses how the present study can be of benefit to psychological research and what possible extensions can be made to overcome its limitations.

Suicide Risk of Bipolar Disorder Patients

Bipolar disorder (BD) is a chronic mood disorder characterized by recurrent manic or hypomanic episodes intermixed with depressive episodes (Miller & Black, 2020). In general, BD is classified into two categories – bipolar I disorder (BD I) and bipolar II disorder (BD II). The Diagnostic and Statistical Manual of Mental Disorders (DSM-5) defines BD I by the presence of at least one episode of mania and BD II by the presence of one episode of hypomania and depression (McCormick et al., 2015). Mania is known to result in more severe functional impairment and psychotic symptoms than hypomania does (Haldane et al., 2008). BD patients are known to have a high risk of dying by suicide (Simpson & Jamison, 1999). In general, it is estimated that 25-60% of individuals diagnosed with BD attempt suicide, and 4-19% die by suicide (Novick et al., 2010). The suicidality of BD patients even exceeds that of patients suffering from other psychiatric disorders (Høyer et al., 2000; Miller & Black, 2020). A comparison study noted that suicide was a great cause of mortality especially in the early stage of BD development with

its ratio of BD patients due to suicide being 23.4 times higher than that of the general population (Sharma & Markar, 1994). Furthermore, it was suggested that BD patients tend to employ more lethal suicide methods than suicide attempters in non-BD population (Dome et al., 2019).

Impulsivity is a long-known important risk factor for suicidality, particularly for repetitive suicide attempt (Gvion et al., 2018). Impulsivity refers to a tendency toward unplanned reactions to stimuli despite predictable negative consequences (Moeller et al., 2001; Najt et al., 2007). BD patients who attempted suicide are believed to possess more severe impulsivity than those who did not. (Moraes et al., 2013; Swann et al., 2005). Impulsivity has been found to be closely related to the deficits in response inhibition (Christodoulou et al., 2006). Response inhibition denotes the ability to control one's thoughts and behaviors to overcome a strong internal predisposition or to thrust external temptation away (Meule, 2017). Deficits in response inhibition is pronounced among BD I patients (APA, 2013). Impaired response inhibition may be linked to an inability to delay reward, eventually resulting in impulsivity (Swann et al., 2009). The common measure of response inhibition is go/no-go task (Verbruggen & Logan, 2008). During go/no-go tasks, participants are asked to respond to the frequent target (go tasks) but to withhold a response to the less frequent target (no-go tasks) by which they get involved with conflict detection (Albanese et al., 2019; Weisbrod et al., 2000). The go/no-go task is a suitable paradigm for investigating response inhibition with event-related potentials (ERPs). ERP is a time-locked measure that represents the changes in the electrical activities of the cerebral surface when stimuli are presented (Patel & Azzam, 2005; Weisbrod et al., 2000). It offers a non-invasive way to examine rapid neural processes underpinning inhibitory control (Albanese et al., 2019). The N200 component is one of the ERPs that have been often investigated in relation to response inhibition (Bokura et al., 2001). N200 is generated under a no-go condition and comprises a negative shift approximately between

200 and 300 ms (Bokura et al., 2001). Multiple ERP studies stated that N200 may be associated with central inhibition or response conflict (Dominke et al., 2021; Gajewski & Falkenstein, 2013; Liotti et al., 2010). There have been mixing implications about the potential of N200 as a neural marker of suicidal risk. A principal component analysis (PCA) on the N200 components collected from suicide attempters and ideating non-attempters showed that the former exhibited a more positively shifted N200 component during a no-go task than the latter (Albanese et al., 2019). However, there was a recent ERP study that found no significant difference in the N200 amplitudes between suicide attempters and non-attempters (Tavakoli et al., 2021).

Suicide Research with Machine Learning

Recent suicide research is actively underway with the applications of ML algorithms. The popularity of ML algorithms is mainly attributed to their learning capability from high-dimensional data and strong predictive power. Previous ML studies showed its promise of detecting individuals with high suicidality not only from the general population but also from the patient groups diagnosed with psychiatric disorders such as depression, bipolar disorder, and schizophrenia (Bohaterewicz et al., 2021; Chen et al., 2020; Fan et al., 2020; Goldstein et al., 2022; Hack et al., 2017; Hasey et al, 2020; Hettige et al., 2017; Miché et al., 2020; Navarro et al., 2021, Passos et al., 2016; Su et al., 2020) (Table 1). The popular ML algorithms to date include support vector machine (SVM), random forest (RF), and logistic regression (LR). Traditional risk factors of high suicidality are demographic variables, previous history of suicide attempt, and self-assessment on suicide-related phenotypes (Bostwick et al, 2016; Franklin et al., 2017). These risk factors, however, have several limitations. First, demographic variables may help recognize differentiated lifetime risk for suicide attempt according to a person's membership but may not

signal sufficiently meaningful patterns for the assessment of person-specific suicidality (Gibb & Tsypes, 2019). Second, using previous suicide attempt as a predictor of suicidal risk may draw biased attention to previous attempters rather than identifying non-attempters who are prone to attempting suicide going forward. Finally, self-assessment by patients can be influenced by their biases (e.g., cultural or experiential) and memory abilities (Gibb & Tsypes, 2019). It may also pose a risk of measurement non-invariance that a psychological construct has a different structure or meaning to different groups or on different measurement occasions in the same group. In this case, the target construct (e.g., suicide risk) cannot be meaningfully tested or construed across groups or across time (Putnick & Bornstein, 2016).

Acknowledging the aforementioned limitations of traditional risk factors, researchers recently began to leverage biological measures such as functional magnetic resonance image (fMRI) and electroencephalography (EEG) data. A resting-state fMRI study predicted suicidal risk of schizophrenia patients with ML algorithms (Bohaterewicz et al., 2021). Also, an EEG study applied ML to resting EEG data in an attempt to detect suicidal ideators from a sample of depression patients (Hasey et al., 2020). However, there is still a paucity of suicide research with ML on biological measures. The aforementioned previous literature on the N200 ERP component suggests its potential as an effectual predictor of suicidal risk. Since decreased response inhibition may be seen in suicide attempters, ERPs captured during response inhibition task can be a meaningful neural marker. To the best of the author's knowledge, no suicide research has been carried out with a combination of ML and ERPs. The current study conducts an empirical analysis with ML and ERPs to predict suicide attempt by BD I disorder patients. The study shows how the initial prediction based on the raw ERP data can be improved by the integrative machine learning approach described herein.

Table 1

Summary of prior machine learning research on predicting suicide risk

Authors	ML	Data	Participants	performance
Bohaterewicz et al. (2021)	boosting; lasso; logistic regression; random forest; support vector machine	multi-state fMRI indexes	N = 59 patients with schizophrenia	70% ACC 76% AUC
Chen et al. (2020)	boosting; logistic regression; random forest;neural network	demographics; criminality	N = 126,205 patients with mental disorders	88-89% AUC 52-75% SNT
Fan et al. (2020)	random forest; logistic regression; k-nearest neighbors; Naïve Bayes	electronic health records	N = 6,042 patients with BD and PTSD	55-98% SNT 7-91% PRC
Goldstein et al. (2022)	boosting	demographics; psychosis; history of suicide attempt	N = 394 BD patients	82% AUC
Hack et al. (2017)	lasso;support vector machine	self-report; clinician assessment	N = 1,017 patients with trauma	71% AUC 64% SNT
Hasey et al (2020)	random forest	resting EEG	N = 40 patients with depression	77% AUC 71% ACC
Hettige et al. (2017)	linear regression; random forest; elastic net regression; support vector machine	sociocultural/clinical data	N = 345 patients with schizophrenia	65-67% ACC 70-71% AUC
Jung et al. (2019)	logistic regression;random forest; neural network; support vector machine, neural network; extreme gradient boosting	risk behavior survey	N = 59,984 middle & high school students	78-79% ACC
Miché et al. (2020)	logistic regression; lasso, ridge, random forest	demographics; clinical records; history of suicide attempt	N = 2,797 a sample from general population	82-83% AUC
Navarro et al. (2021)	random forest	pregnancy/birth record; parent assessment	N = 1,623 a cohort for a longitudinal study	62-72% AUC
Passos et al. (2016)	lasso; support vector machine; relevance vector machine	clinical records; demographics; history of trauma	N = 144 patients with mood disorders	67-73% ACC
Su et al. (2020)	penalized logistic regression	demographics; medications; laboratory test; diagnosis codes	N = 41,721 patients with clinical records	81-88% AUC 51-65% SNT 90-95% SPC

Note. AUC: area under the curve, ACC: accuracy, SNT: sensitivity, SPC: specificity, PRC: precision

CHAPTER 2: EMPIRICAL DATA

Participants

Fifty-seven individuals diagnosed with BD I participated in the study. These individuals were drawn from the Pretcher Bipolar Research Program (McInnis et al., 2018). Thirty-five participants had no history of suicide attempt (NSA), while twenty-two of them had attempted suicide at least once (SA) in their life time. The Diagnostic Interview for Genetic Studies (DIGS) was conducted by a trained clinician, and the diagnosis of BD I was confirmed by two doctorate level reviewers (Nurnberger et al., 1994). The primary language of all the participants was English, and their vision was normal or corrected to normal. All the participants in the study were right-handed. The two groups were matched on age, gender, and education (Table 2).

Table 2

Demographic characteristics of participants

	NSA	SA	Test	*p*-value	CI
Age (years)	41.34 (12.18)	42.23 (7.77)	$t(54.97) = .33$	.74	(-4.41, 6.18)
Education (years)	15.58 (2.39)	15.23 (2.67)	$t(41.55) = -.49$	.62	(-1.77, 1.07)
Gender (male/female)	18 / 17	8 / 14	$\chi^2(1) = .703$	.40	-

Note. Age, Education: mean (standard deviation)

There were two missing values in the years of education.

Visual Go/no-go Tasks

The electroencephalograms (EEGs) were recorded during a modified version of a visual go/no-go task (Eimer, 1993). Visual stimuli were generated using the Corel Photopaint 6.0 graphics program. The central plus sign or central arrow, subtending 1 degree of visual angle (deg), was flanked by two squares, each of which subtended 2 deg of visual angles. Each square was 6 deg to the left and right on the horizontal meridian. Two alphabet letters (M and W), subtending 1 degree each, were presented at the center of either a right or left square. The task consisted of 240 trials presented in 4 separate blocks of 60 trials each, in which the go:no-go ratio was 7:3. In two blocks (120 trials) the letter M and W were the go and the no-go stimulus respectively, and the vice versa for the other two blocks. All participants were instructed to press the right or left button as fast as possible when the go stimulus appeared on the right or left side and to withhold response when the no-go stimulus was presented on either side. The order of four blocks were counterbalanced across the participants to control any order effects. Each trial was preceded by an arrow as a precue (200 ms) with 100% validity, indicating on which side a stimulus would be presented, followed by an inter-stimulus interval (700 ms) and a target (M or W, 150 ms).

ERP Preprocessing

Data were processed with software developed by the James Long Company. ERPs from 9 electrode sites, including F3, Fz, F4, C3, Cz, C4, P3, Pz, and P4, were analyzed, and only correct trials were included (Chun et al., 2013). Artifacts due to eye blinks were corrected by linear regression (Gratton et al., 1983). EEG data were divided into the two segments of 150 ms pre-

stimulus onset and 1000 ms post-stimulus onset. The data were baseline corrected before averaging. All trials with remaining artifacts were excluded from further analyses.

N200 Peak Latency ERPs

ERPs are scalp-recorded voltage fluctuations that are time-locked to an event, obtained by averaging EEG fragments across multiple trials (Kropotov, 2016). Per participant, 250 data points of averaged time-stamped ERP amplitudes were collected from 9 electrode sites every 4ms. Temporal principal component analysis (tPCA) was implemented to the time-series data with Varimax rotation to decompose ERPs into a discrete set of temporal patterns (Chapman & McCrary, 1995; Coles & Gratton, 1986; Donchin & Coles, 1988; Frishkoff et al., 2007). tPCA[1] calculates the covariance between all ERP time points and forms groups of highly covarying time points that constitute individual ERP components (Albert et al., 2012; Scharf et al., 2022). Among several candidate factor models, 8-factor model was chosen based on the best fit to the grand averaged ERP amplitudes for all participants. From this model, the time window of N200 was identified between 150-308 ms among BD I patients. Thus, the N200 peak latency is defined as the time when the most negative amplitude was observed between 150-308 ms post-stimulus onset time window. ERPs were measured from 9 electrode sites, 3 (frontal, central, parietal region) × 3

[1] Much ERP literature has called a statistical method used in decomposing an observed brainwave signal into a set of underlying latent constructs as tPCA. However, what it actually implemented is factor analysis on the voltage (t_{ij}) collected from an observation i and a sampling point j. Scharf et al. (2022) noted that factors refer to the estimates of the true underlying signals, while components refer to the ideally recovered signal. Albeit similarities between PCA and factor analysis (e.g., their utility for dimension reduction and the involvement of latent constructs in modeling), they are completely distinct given that PCA predicts latent constructs using a set of observed independent variables, while factor analysis predicts observed variables with a set of latent constructs. Since the term, tPCA, has been widely used in this field, the identical term was used here, but tPCA does not denote PCA generally used in the realm of statistics.

(left, midline, right side), while 2 types of stimuli (Go, NoGo) were presented to 2 visual fields (right and left). As a result, N200 peak latency ERP data consists of 36 variables (3 × 3 × 2 × 2).

Variables

Predictor variables consist of thirty-six N200 ERP peak latencies (Table 3). A response variable is binary as to history of suicide attempt at baseline visit (SA: suicide attempter, NSA: suicide non-attempter). All the values of predictor variables were Z-transformed. Then, participants that had at least one absolute value of transformed latencies greater than 3 were considered as outliers. As a result, four participants classified as NSA were removed from the final dataset. Therefore, the following analyses are continued with 53 observations (NSA - 31, SA - 22).

Table 3

Predictor variables

Laterality / Caudality	3 (left)	z (midline)	4 (right)
F (front)	F3S1, F3S2, F3S3, F3S4	FzS1, FzS2, FzS3, FzS4	F4S1, F4S2, F4S3, F4S4
C (central)	C3S1, C3S2, C3S3, C3S4	CzS1, CzS2, CzS3, CzS4	C4S1, C4S2, C4S3, C4S4
P (parietal)	P3S1, P3S2, P3S3, P3S4	PzS1, PzS2, PzS3, PzS4	P4S1, P4S2, P4S3, P4S4

Note. S1: Go stimulus presented to the right visual hemifield (RVF)

S2: Go stimulus presented to the left visual hemifield (LVF)

S3: NoGo stimulus presented to the LVF

S4: NoGo stimulus presented to RVF

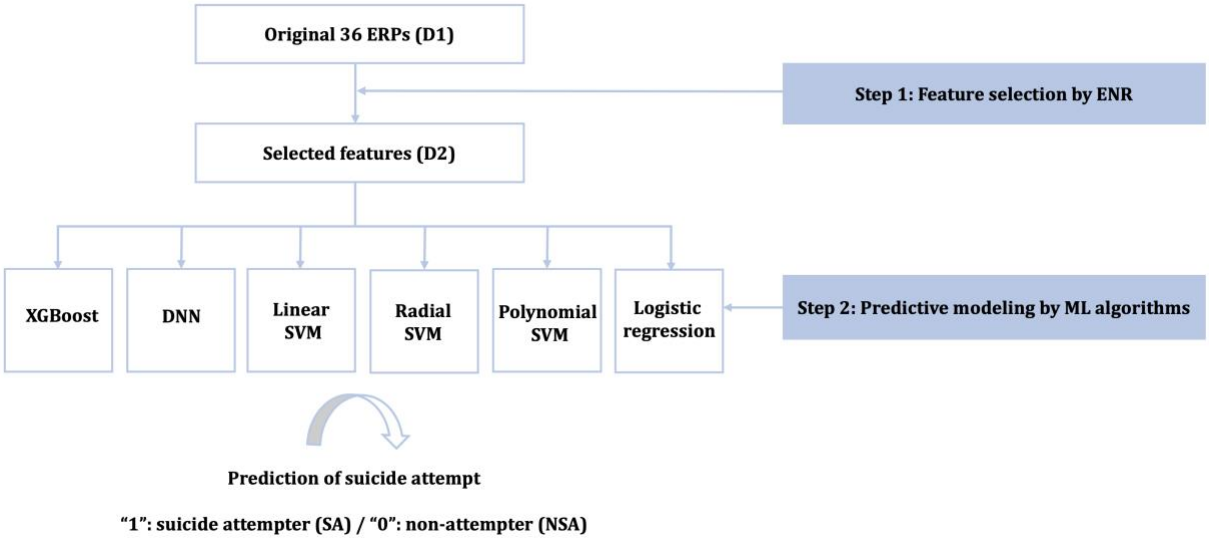
CHAPTER 3: METHODS

Analytical Procedures

The current paper demonstrates three analyses on predicting suicide attempt of BD I patients by classification supervised ML, using their N200 ERP peak latencies (ERPs). Analysis 1 was designed to address high dimensionality in the small sample. Here we explore the efficacy of feature selection in advance of predictive modeling (Figure 1). First, elastic net logistic regression (ENR) is performed on 36 original ERPs (D1) in an attempt to select important features in the relationship with suicide attempt, reducing the dimensionality of predictors. The selected features are referred to as D2. Second, predictive modeling is run by six supervised ML algorithms – support vector machine with linear, radial, and polynomial kernels, deep neural network, extreme gradient boosting, and logistic regression. Then, their predictive performances on D2 are compared to those on D1. Analysis 2 implements the integrative ML approach as a way of further improving prediction shown in analysis 1. This analysis leverages the oversampled features (D3) for classification (Figure 2). In creating D3, a minority oversampling technique (SMOTE) is embedded between feature selection and predictive modeling and generates new minority instances based on the pre-existing ones to resolve class imbalance. The prediction outcomes on D2 are compared to those on D3, and the usefulness of SMOTE is explored. Analysis 3 examines the utility of ERPs as the potential neural marker of suicide attempt. Here the prediction results on D2 are compared to those with demographic variables (D4). Across three analyses, supervised learning algorithms implement leave-one-out cross-validation to fully exploit the small sample.

Figure 1

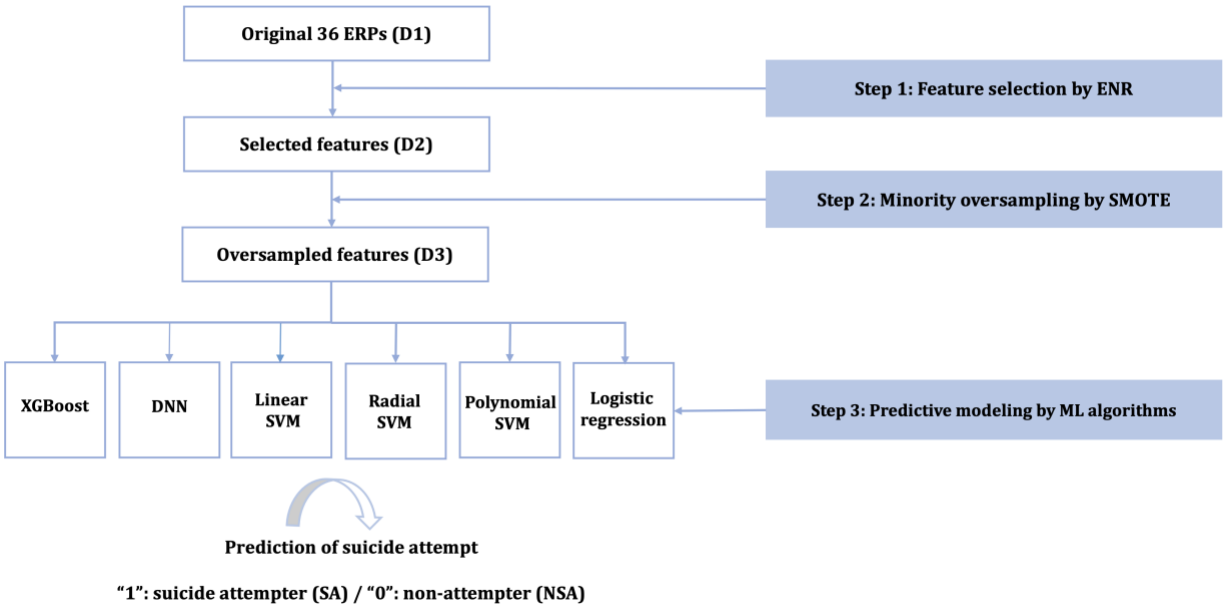
Representation of analysis 1: feature selection + predictive modeling



Note. Analysis 1 consists of two steps – feature selection by ENR and predictive modeling by ML algorithms. Important features in relation to suicide attempt are extracted via feature selection and are sequentially fed into predictive modeling by six supervised ML algorithms. The trained models predict the suicidality of BD I patients via binary classification.

Figure 2

Representation of analysis 2: integrative classification approach



Note. Analysis 2 consists of three steps – feature selection by ENR, minority oversampling by SMOTE, and predictive modeling by supervised ML algorithms. Analysis 2 embeds the minority oversampling process between feature selection and predictive modeling. The oversampled final features contain the equal number of observations for each class.

Machine Learning Algorithms

Overview

Machine learning (ML) refers to an application of artificial intelligence, aiming to enable computers to learn from given data so as to generate statistical models that perform data-driven analyses for making predictions or inferring relationships among variables (Bi et al., 2019; James et al., 2013; Mahesh, 2018). In general, ML is classified into three categories – supervised, semi-supervised, and unsupervised learning. Given that the current research implements supervised learning only, this section will closely look into supervised ML algorithms.

A supervised ML algorithm builds a statistical model for the purpose of prediction (James et al., 2013). The algorithm fits the model on the pre-defined set of training data of which labels are known. Then, it leverages knowledge accumulated throughout training process to predict the labels of test data of which labels are assumed to be unknown. Comparing the predicted labels and the actual labels of test data, the performance of the predictive model is evaluated. When labels are categorical, supervised learning performs classification, or otherwise, it runs regression. This study performs classification to construct predictive models (i.e., classifiers) that differentiate suicide attempters (SA) from non-attempters (NSA), using N200 ERP peak latencies drawn from a sample of BD I patients. Six ML algorithms are implemented for predictive modeling, some of which have been at the forefront (e.g., support vector machine and logistic regression) and others of which have yet been often employed in suicide research (e.g., deep neural network and XGBoost). Most of the algorithms contain a set of hyperparameters that largely affect the performance of classifiers. In tuning hyperparameters, random search and grid search were performed in a consecutive manner. Random search was first conducted to uncover the

neighborhood which the optimal values are likely to lie in. Then, grid search was followed to scrutinize the identified area so as to pick the optimal values of hyperparameters. Across both tuning processes, leave-one-out cross-validation was implemented.

ML Algorithms for Predictive Modeling

Support Vector Machine. Support vector machine (SVM) searches for the optimal separating hyperplane farthest from observations (James et al., 2013). A hyperplane is a flat affine subspace of $p - 1$ dimension in the presence of p predictors (James et al., 2013). SVM aims to maximize margin, the minimum perpendicular distance from the observations to the hyperplane. Support vectors are the observations that fall on the margin, or on the wrong side of the margin for their class. If training data are linearly separable, SVM with linear kernel will satisfy (1).

$$\forall i = 1,...,N,\ M > 0,\ \ \varepsilon_i \geq 0,\ \sum_{i=1}^{N} \varepsilon_i \leq C,\ \ y_i \in \{-1, 1\} \tag{1}$$

$$maximize_{\beta_0, \beta_{1,...,} \beta_p, \varepsilon_{1,...,} \varepsilon_N} M$$

$$subject\ to \sum_{j=1}^{p} \beta_j^2 = 1$$

$$y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ ... \ + \beta_p x_{ip}\right) \geq M\left(1 - \varepsilon_i\right)$$

$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ ... \ + \beta_p x_{ip} = 0$ defines a hyperplane. M is the width of the margin, and ϵ_i is a slack variable that dictates the degree of individual observations being allowed to lie on the wrong side of the margin. C is a budget for the amount that the margin can be violated by observations, in other words, the degree of tolerance to violations to the margin (James et al., 2013). In building SVM classifiers, the *cost* argument was tuned instead of C. The *cost* argument and C

are not the identical terms but in an inverse relationship as the former literally denotes the cost of violations to the margin.

The advantage of using SVM is that we can achieve not only a linear classifier but also non-linear classifiers depending on the choice of kernels. A kernel function, $K < \boldsymbol{x}, \boldsymbol{x}' >$, is a symmetric, positive (semi-) definite function that represents the transformation of input data by the inner product of two input vectors given the number of predictors, p (2) (Hofmann et al., 2008; James et al., 2013). Equations (3) to (5) denote three kernel functions – linear kernel (3), polynomial kernel with d-th degree (4), and radial basis kernel with gamma (γ) (5).

$$\boldsymbol{x}_1 = (x_{11}, \ldots, x_{1p})^T, \ \boldsymbol{x}_2 = (x_{21}, \ldots, x_{2p})^T$$

$$< \boldsymbol{x}_1, \boldsymbol{x}_2 > \ = \ \boldsymbol{x}_1{}^T \boldsymbol{x}_2 \tag{2}$$

$$K < \boldsymbol{x}_1, \boldsymbol{x}_2 > \ = \ \boldsymbol{x}_1{}^T \boldsymbol{x}_2 \tag{3}$$

$$K < \boldsymbol{x}_1, \boldsymbol{x}_2 > \ = \ \left(\boldsymbol{x}_1{}^T \boldsymbol{x}_2 + 1 \right)^d \tag{4}$$

$$K < \boldsymbol{x}_1, \boldsymbol{x}_2 > \ = \ exp\left(-\gamma \, \| \boldsymbol{x}_1, \boldsymbol{x}_2 \|^2 \right) \tag{5}$$

Three types of SVM models were generated by linear, radial basis, and polynomial kernels in the study. The *tune* function and the *svm* function embedded in the R library *e1071* were used for tuning the hyperparameters of each SVM model and for building SVM models on the basis of the tuned hyperparameters (Meyer et al., 2021). The hyperparameters of three SVM models are listed in Table 4.

Table 4

Main hyperparameters of SVM

Hyperparameters	Description	Relevant kernel
cost	controls the trade-off between the number of slack variables and the width of the margin	linear, radial, polynomial
gamma	dictates the extent to which a decision boundary curves	radial, polynomial
degree	determines the degree of polynomial function	polynomial

Logistic Regression. Logistic regression generates a linear classifier with respect to log odds ratio. It computes the posterior probabilities for each class per observation and assigns it to a class for which the posterior probability is the highest (Maalouf, 2011). Let $Pr(\boldsymbol{x}) = Pr(Y = 1|X = \boldsymbol{x})$ and $1 - Pr(\boldsymbol{x}) = Pr(Y = 0|X = \boldsymbol{x})$. Pr is the probability function with the range [0,1]. With the logit function g defined as (6), the range of $g(\Pr(\boldsymbol{x}))$ is shifted to (-∞, ∞).

$$g(Pr(\boldsymbol{x})) = ln\frac{Pr(Y=1|X=\boldsymbol{x})}{Pr(Y=0|X=\boldsymbol{x})} = ln\left\{\frac{Pr(\boldsymbol{x})}{1-Pr(\boldsymbol{x})}\right\} = \boldsymbol{x}^T\boldsymbol{\beta} \tag{6}$$

$$Pr(\boldsymbol{x}) = \frac{e^{\boldsymbol{x}^T\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}^T\boldsymbol{\beta}}};\ 1-Pr(\boldsymbol{x}) = \frac{1}{1 + e^{\boldsymbol{x}^T\boldsymbol{\beta}}};\ \boldsymbol{x} = \left\{1, x_1,\ldots, x_p\right\}^T;\ \boldsymbol{\beta} = \left\{\beta_0, \beta_1,\ldots, \beta_p\right\}^T$$

Here, $\boldsymbol{x}$ denotes a $(p + 1) \times 1$ column vector for an observation where p is the number of predictors. We search for the coefficient estimates $\boldsymbol{\beta}$ maximizing the log-likelihood function, $l(\boldsymbol{\beta})$, with respect to N observations in (7).

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{N} \ln\left[Pr\left(\boldsymbol{x}_i\right)^{y_i}\left\{1 - Pr\left(\boldsymbol{x}_i\right)\right\}^{1-y_i}\right] \quad y_i \in \{0,1\} \tag{7}$$

$$= \sum_{i=1}^{N} \left[y_i \ln Pr(\boldsymbol{x}_i) + (1 - y_i) \ln \left\{ 1 - Pr(\boldsymbol{x}_i) \right\} \right]$$

$$= \sum_{i=1}^{N} \left[y_i \ln \frac{Pr(\boldsymbol{x}_i)}{1 - Pr(\boldsymbol{x}_i)} + \ln \left\{ 1 - Pr(\boldsymbol{x}_i) \right\} \right]$$

$$= \sum_{i=1}^{N} \left\{ y_i \boldsymbol{x}_i^T \boldsymbol{\beta} - \ln \left(1 + e^{\boldsymbol{x}_i^T \beta} \right) \right\}$$

We take derivatives on the likelihood with respect to $\boldsymbol{\beta}$ and set them equal to zero to attain the maximum log-likelihood (8). This process does not give rise to a solution in a closed form, so the iteratively reweighted least squares algorithm is used to estimate the maximum likelihood estimates of $\boldsymbol{\beta}$. Logistic regression is simple to implement with no hyperparameters and shows the importance of predictors (i.e., regression coefficients) and their directed causality with a sign (+ or −). Logistic regression models were built by the *glm* function in the *stats* R library (R Core Team, 2013). The *family* argument was set to *binomial* to perform binary classification.

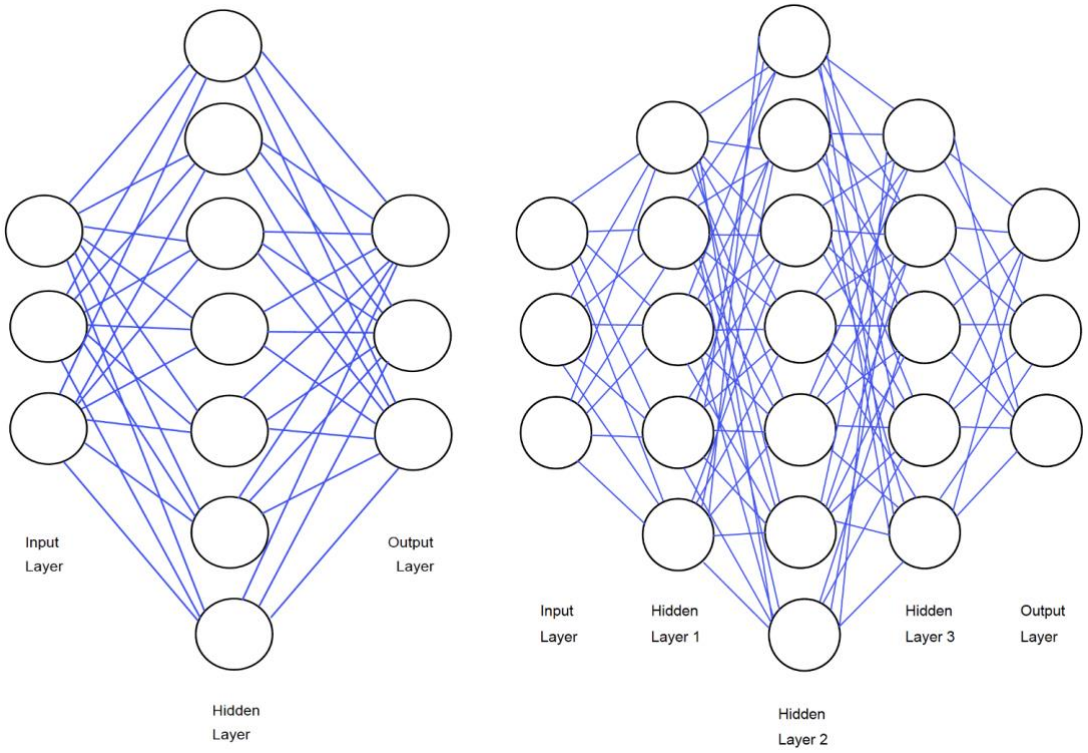
$$\frac{\partial l(\boldsymbol{\beta})}{d\boldsymbol{\beta}} = \sum_{i=1}^{N} \left\{ y_i \boldsymbol{x}_i^T - \frac{\boldsymbol{x}_i^T e^{\boldsymbol{x}_i^T \beta}}{1 + e^{\boldsymbol{x}_i^T \beta}} \right\} \tag{8}$$

Deep Neural Network. Neural network is a powerful supervised learning algorithm with networks of multiple nodes of neurons (Hastie et al., 2017; Sharma et al., 2017) (Figure 3). Neural network mimics the way the human brain learns through its neuron network. It consists of three layers – input, hidden, and output layer. The input layer is the initial layer that holds input data and starts disseminating them via synapses into the nodes situated in the following hidden layer. The hidden layer bridges the input layer and the output layer. The output layer is the final layer that

receives input from the previous hidden layer. Each node in the hidden layer and the output layer has an activation function that determines whether to turn the node on or off.

Figure 3

Illustrated structures of NN and DNN



Note. (Left) A neural network consists of an input layer, a hidden layer, and an output layer. A circle indicates a node, and a blue straight line portrays a synapse that connects two nodes in adjacent layers. (Right) A deep neural network (DNN) shows more intricately connected network than the neural network with more than one hidden layer. This illustration shows a DNN with three hidden layers. In both pictures, we can observe three nodes in the output layer, meaning that those networks are supposed to perform three-class classification. This study implements binary classification to differentiate suicide attempters from non-attempters. Therefore, there are two nodes in the output layer of our DNN models.

Deep neural network (DNN) has more sophisticated architectures than neural network with more than one hidden layer. Such intricacy renders DNN to be more powerful than neural network in catching complex relationships between inputs and outputs (Srivastava et al., 2014). Supervised learning by DNN is processed throughout multiple roundtrips between the input and the output layer. This back-and-forth ping-pong is run by forward and backward propagation. In forward propagation, input flows from the input to the output layer, while in backward propagation, learning flows in reverse, modifying the parameters estimated during the prior forward propagation. Throughout the entire learning process, DNN seeks for the optimal values of two parameters – weight ($\boldsymbol{w}$) and bias (c). Weight controls the strength of signal between two nodes connected via synapses. Bias is a constant term added to the weighted sum of input. When an input ($\boldsymbol{x}_i$) is transmitted via synapses to the next layer, it is newly weighted, and the weighted sum of input is added to bias (c) ($\sum_{i=1}^{n} \boldsymbol{w}_i \boldsymbol{x}_i + c$), which sequentially enters an activation function (f) (9).

$$\sum_{i=1}^{N} \boldsymbol{w}_i \boldsymbol{x}_i + c \rightarrow f\left(\sum_{i=1}^{N} \boldsymbol{w}_i \boldsymbol{x}_i + c\right) \tag{9}$$

Activation functions introduce non-linearity to the network and largely affect its prediction performance (Sharma et al., 2017). The sum of weighted input and bias becomes the domain of an activation function. Weight reflects the slope of an activation function, while bias exerts influence on the movement of an activation function from side to side. There are multiple types of activation functions (Appendix 1). This study used ReLU function in the hidden layers and Softmax function in the output layer. ReLU function has been the most popular activation function to date in deep learning research mainly due to its sparsity like biological neurons that less than 5% of neurons

are activated simultaneously (10) (Ding et al., 2018; Glorot & Bengjo, 2018; Krizhevsky et al., 2017 ; Srivastava et al., 2013; Zeiler et al., 2013). ReLu function operates elementwise, vanishing negative elements but returning the original values for non-negative elements.

$$f(x) = \max(0, x) \tag{10}$$

Softmax function has been the dominant activation function used in the output layer of neural network (11) (Nwankpa et al., 2018). $f(\boldsymbol{x})_j$ represents the probability of an input vector $\boldsymbol{x}$ to be categorized into the class j among Q number of classes. Softmax converts the outputs into the probability values summed up to 1. The output layer was designed to have two nodes, each of which represents a class, with one-hot encoding that assigns binary classes into two corresponding columns.

$$f(\boldsymbol{x})_j = \frac{e^{\boldsymbol{x}_j}}{\sum_{q=1}^{Q} e^{\boldsymbol{x}_q}} \ (j = 1,...,Q) \tag{11}$$

Now we turn to the optimization methods for DNN. Optimization refers to the process in search of the optimal values of parameters that locally minimizes loss function. Stochastic gradient descent algorithm (SGD) underlies the majority of optimization methods for DNN (Zhang, 2018). SGD updates parameters (i.e., weight and bias) by subtracting the initial parameters by the product of learning rate and gradient (12). L is loss function; w_i refers to the weight of an input $\boldsymbol{x}_i$; c is bias; η is learning rate; $\frac{\partial L}{\partial w_i}$ and $\frac{\partial L}{\partial c}$ are gradients.

$$w_i^* = w_i - \eta * \frac{\partial L}{\partial w_i} \; ; \; c^* = c - \eta * \frac{\partial L}{\partial c} \tag{12}$$

SGD starts from randomly set parameters and travels on its terrain until it reaches the minimum point of loss function. Even though SGD expedites the entire learning process, using a small subset of input data (i.e., batch) throughout optimization process, it has several disadvantages. First, it draws inefficient zigzag paths through iterative learning. Second, optimization sometimes discontinues at a local minimum or a saddle point where gradient is zero. Third, SGD assigns a global learning rate across all parameters despite their varied ranges.

Multiple variants of SGD such as Adagrad, RMSprop, Momentum, and Adam were devised in an attempt to overcome the aforementioned limitations (Zhang, 2018). Adagrad (adaptive gradient algorithm) and RMSProp (root mean squared propagation) implements an adaptive learning rate that varies for every parameter at each iteration rather than using the global learning rate across the entire learning processes for all the parameters. Adagrad takes a large learning step when reaching sparse features but moves slowly around dense features to scrutinize their neighborhood (Lydia & Francis, 2019). The shortcoming of Adagrad is that the learning rate keeps decreasing in each iteration (Septiadi et al., 2020). To tackle this issue, RMSprop introduces the moving average of gradients so that the optimizer focuses more heavily on the recent gradient but less heavily on the old gradients. Momentum controls not only the learning speed but also the direction of learning (Nakerst et al., 2020). It increases the momentum by inertia when gradients keep moving to the same direction. Therefore, even when the optimizer reaches a local minimum or a saddle point, Momentum can keep moving forward and escape from those flat points. Adam (adaptive moment estimation) is the most popular optimizer in deep learning, which borrows the merits of RMSProp and Momentum. Adam controls both the size of learning step and the direction

of learning so that the optimizer can learn enough from each observation without being too rapidly decayed nor inefficiently taking a zigzag path. This study uses Adam as the optimizer for DNN. In terms of loss function, the categorical cross-entropy was used with one-hot encoding. Categorical cross-entropy is used for classification and generates probabilities that an observation belongs to each class, q, that are summed up to 1 (13). Cross-entropy loss tends to increase when predicted probabilities deviate from the actual class.

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{y_{iq}\log\left(Pr_{iq}\right) + \left(1-y_{iq}\right)\log\left(1-Pr_{iq}\right)\right\} \quad (13)$$

The power of DNN is mainly attributed to multiple processing layers where activation functions are situated, which effectively detect complex non-linear relationships among variables. However, DNN has several limitations. First, DNN is a black box in nature. That is, it does not reveal how the model works internally in producing output. Second, DNN tends to show excellent performance when sample size is large. So, we cannot guarantee that DNN would still demonstrate high performance given small samples. Finally, DNN is prone to overfitting, largely due to its requirement of large data and its sheer capacity to memorize massive data (Zhang & Sabuncu, 2018). Multiple ways to avoid overfitting have been introduced. First, limiting the number of hidden layers and hidden units can be considered. Second, weight decay is a useful L2 regularization method that constrains weights from growing too fast (Zhang et al., 2018). Third, we can consider stopping learning early by limiting the number of roundtrips (i.e., epoch). Fourth, we may use the dropout method that excludes partial nodes from learning process by introducing the proportion of retaining nodes (Srivastava et al., 2014). Fifth, we can control the batch size, a

number of samples to be used at each update of model parameters. A smaller batch size has been known to better generalize a trained model (Hoffer et al., 2017).

The current study used the R libraries *keras* and *tensorflow* to build DNN models (Falbel et al., 2021; Kalinowski et al., 2021). Considering a large number of hyperparameters, the architecture of DNN was designed in three steps. First, the number of hidden layers and the number of nodes in each layer were determined. Too many hidden layers and nodes were avoided to prevent overfitting. Second, activation functions, loss function, and optimizer were chosen as aforementioned with reference to prior deep learning literature. Finally, the remaining hyperparameters were tuned by LOOCV. The hyperparameters of DNN are shown in Table 5.

Table 5

Main hyperparameters of DNN

Hyperparameters	Description
number of hidden layers	a higher number of hidden layers leading to a more complex model
number of nodes	a higher number of hidden units leading to a more complex model
learning rate	step size at each iteration
dropout	probability at which outputs of the layer are dropped out
batch size	the proportion of samples used at each update of model parameters
activation function	function that determines whether a neuron node is activated
weight decay	a value multiplied to weights to prevent them from growing too fast
epoch	number of roundtrips with a pair of forward pass and backward pass
optimizer	algorithm optimizing model parameters until reaching the minimum loss
loss function	objective function that optimizer attempts to minimize

Extreme Gradient Boosting. Extreme gradient boosting (XGBoost) is an ensemble algorithm that integrates gradient boosting framework into a base learner to generate a unified predictive model (Chen et al., 2021). A base learner forms the basic architecture of XGBoost. Gradient boosting gradually improves the predictive power of a weak base learner in an additive manner. Here, a weak learner function, $h(\boldsymbol{x}, \boldsymbol{\theta})$, refers to a classifier of which prediction accuracy is barely above random guess (50% of accuracy). An additive manner means that a boosting algorithm iteratively fits $h(\boldsymbol{x}, \boldsymbol{\theta})$ to the pseudo-residual generated from the previous round of learning. Given the b-th iteration $(b = 1, \ldots, B)$ and an observation i $(i = 1, \ldots, N)$, the pseudo-residual ($\tilde{y}_{ib}$), the difference between an actual value and a predicted value, serves as the negative gradient of the loss function $\Psi\left(y_i, \hat{f}(\boldsymbol{x}_i)\right)$ to be minimized (14) (Friedman, 2002). In binary classification, $\Psi\left(y_i, \hat{f}(\boldsymbol{x}_i)\right)$ is often the negative log-likelihood (see equation 7).

$$\tilde{y}_{ib} = -\left[\frac{\partial \Psi\left(y_i, \hat{f}\left(\boldsymbol{x}_i\right)\right)}{\partial f\left(\boldsymbol{x}_i\right)}\right]_{\hat{f}(\boldsymbol{x}) = \hat{f}_{b-1}(\boldsymbol{x})} \tag{14}$$

ρ_b, the optimal step length or the magnitude of contribution that an individual base learner makes to the final predictive model, is estimated at the b-th iteration (15), and the predictive model is sequentially updated as (16).

$$\varrho_b = \underset{\varrho}{argmin} \sum_{i=1}^{N} \Psi\left\{y_i, \hat{f}_{b-1}\left(\boldsymbol{x}_i\right) + \varrho h\left(\boldsymbol{x}_i, \boldsymbol{\theta}_b\right)\right\} \tag{15}$$

$$\hat{f}_b(\boldsymbol{x}) \leftarrow \hat{f}_{b-1}(\boldsymbol{x}) + \varrho_b h\left(\boldsymbol{x}, \boldsymbol{\theta}_b\right) \tag{16}$$

XGBoost is an advanced derivative of gradient boosting. XGBoost counters overfitting by incorporating the L1 and L2 regularization terms into loss function. And it takes a multi-threaded approach that speeds up computation with multicore parallel processing. Therefore, unlike gradient boosting that adds a weak learner at a time after the other, the updates of weak learners occur almost simultaneously in XGBoost (Ramraj et al., 2016). XGBoost also employs the sparsity-aware split finding algorithm by which sparsity in data is automatically handled. In addition, it can be a highly customized algorithm in the sense that evaluation metrics and loss function can be chosen according to a researcher's preference.

The current study used the *xgboost* function embedded in the *xgboost* R library (Chen et al., 2021). Either a tree or a linear model could be used as the base learner, but the ERP data showed a better fit with the linear booster. Therefore, XGBoost algorithm generated classifiers with the *gblinear* booster in the study, tuning its hyperparameters – *lambda*, *alpha*, and *nrounds* (Table 6).

Table 6

Main hyperparameters of XGBoost

Hyperparameters	Description
booster	gblinear
nrounds	number of boosting iterations
lambda	L2 regularization term on weights
alpha	L1 regularization term on weights

ML Algorithm for Feature Selection

Elastic Net Logistic Regression. The original ERP data have too many variables ($p = 36$) in the small sample ($N = 53$). Also, those variables tend to be correlated with each other (e.g., ERPs collected from the same electrode scalp sites). A large number of variables in a small sample increase the complexity of data, which renders the fitting of predictive models to be unstable, in other words, overfitting (Bickel et al., 2006; Breiman, 1996). Feature selection methods can help prevent overfitting by removing non-representative predictors while reducing the dimensionality of data. The current study performed feature selection with elastic net logistic regression (ENR). Those who only seek for reducing dimensionality of data rather than selecting representative predictors can consider using dimension shrinkage methods such as principal component analysis (PCA) that extracts features as a combination of original predictors.

Elastic net regression is a compromise between $L1$ regularization and $L2$ regularization. In both regularization methods, a positive regularization parameter λ controls the degree of coefficient shrinkage. The least absolute shrinkage and selection operator regression (i.e., lasso) uses the $L1$ regularization technique. Lasso regression adds the $L1$ penalty term – the penalized summed magnitude of coefficients – to the loss function of ordinary least squares regression (17). Lasso shrinks the coefficient estimates of non-significant predictors towards zeros. In this way, less important features are removed, while more important features remain in the model. Such a property enables lasso to perform feature selection.

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{LASSO}} = arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\left|\beta_j\right| \tag{17}$$

Meanwhile, ridge regression uses the $L2$ regularization technique. Ridge adds the $L2$ penalty term – the penalized summed squares of coefficients – to the loss function of ordinary least squares regression. Ridge regression minimizes the coefficients of non-representative predictors and keeps correlated predictors together in the model (18) (Zou & Hastie, 2005).

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{RIDGE}} = arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\beta_j{}^2 \tag{18}$$

Both lasso and ridge have limitations in practice. First, lasso may not be an effective algorithm to select features in the presence of high collinearities among predictors. When there is a group of predictors highly correlated, lasso tends to randomly choose only one out of the group and removes the rest (Freijeiro-González et al., 2022; Zou & Hastie, 2005). In addition, lasso selects at most n predictors when $p > N$ (Zou & Hastie, 2005). Second, ridge does not perform feature selection but only minimizes marginal coefficients, not shrinking them towards zeros. Elastic net regression borrows the benefits from both lasso and ridge. It selects features like lasso but keeps highly correlated coefficients in the model like ridge. Thus, elastic net regression enables a set of correlated predictors to co-exist but produces a parsimonious selection of predictors (19).

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{ELASTIC\ NET}} = arg\min_{\boldsymbol{\beta}}\left\{\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\left(\alpha\sum_{j=1}^{p}\left|\beta_j\right| + \frac{1-\alpha}{2}\sum_{j=1}^{p}\beta_j{}^2\right)\right\} \tag{19}$$

α controls the relative balance between the lasso and the ridge regression (lasso: α =1, ridge: α = 0). λ adjusts the overall intensity of penalization. As this study predicts a binary response variable, we perform elastic net logistic regression (ENR). The equation (19) is the solution with respect to

linear regression, not logistic regression. Therefore, the loss function of logistic regression (i.e., negative log-likelihood) is plugged into (19) to generate the ENR solution reformulated as (20).

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{ENR}} = arg\min_{\boldsymbol{\beta}}\left[-\sum_{i=1}^{N}\left\{y_i \boldsymbol{x}_i^T\boldsymbol{\beta} - ln\left(1 + e^{\boldsymbol{x}_i^T\boldsymbol{\beta}}\right)\right\} + \lambda\left(\alpha\sum_{j=1}^{p}\left|\beta_j\right| + \frac{1-\alpha}{2}\sum_{j=1}^{p}\beta_j{}^2\right)\right] \quad (20)$$

The present study used the *cva.glmnet* function in the R *glmnetUtils* library to simultaneously cross-validate α and λ, seeking for their optimal values (Friedman et al., 2017; Microsoft & Ooi, 2021). This function originates from the *glmnet* function that fits generalized linear models or regularization models. Its arguments were set as *family* = *binomial* and *type.measure* = *class*. *family* specifies the type of a probability distribution. This study used *binomial* distribution to run ENR, which is the penalized logistic regression with a response variable being binary. The *type.measure* = *class* specifies misclassification error as the loss function for cross-validation (Hastie & Qian, 2016).

Essential Statistical Methods

This section describes leave-one-out cross-validation (LOOCV) and synthetic minority oversampling technique (SMOTE). Two methods are effective for addressing small sample size and class imbalance. The empirical ERP data to be studied has small sample size (N = 53) and a slight class imbalance in the categories of observations (NSA : SA ≒ 3:2). Suicide research mainly aims at detecting those who have high suicidality, but the dominant class in the current ERP data is the low-risk group. Such an unequal composition of classes may not be desirable in the sense that ML algorithms would have more chances to learn from observations that belong to the low

suicidality group and thus predict them better than the high suicidality group. Even though the two classes are not imbalanced as severely as 5:1 or 10:1, the present study takes advantage of this example for demonstrating how to address class imbalance in machine learning when needed.

Leave-one-out Cross-validation

Cross-validation contributes to estimating a classifier's performance and tuning the hyperparameters of ML algorithms. It randomly splits data into two disjoint parts – training data and test data (Zhang & Yang, 2015). Training data are used for model estimation, and test data are used for model validation. Typically, at each round of cross-validation, 75-90% of the sample serves as training data, while the remaining portion is used as test data. However, such splits may be sub-optimal for small samples as we get to train models with too few numbers of observations at each round of cross-validation. To address this issue, this study implements leave-one-out cross-validation (LOOCV) for supervised learning. LOOCV holds out one observation at a time as a test set, leverages the remaining portion of the observations as a training set, and repeats the split process across N times. Below describes the overall process of LOOCV.

Leave-one-out cross-validation (LOOCV)
1. Split the entire data set (N) into a training set ($N-1$) and a test set (1)
2. Fit a predictive model using the training set
3. Validate the model with the test set
4. Repeat step 1-3 N times with a different observation being a test set at a time
5. Collect all the validated outcomes and evaluate the performance of the predictive models

This study employed LOOCV for two reasons. First, it helps secure the largest number of training set and utilize the given sample to the fullest when sample size is small. A limited number of observations in training set tends to compromise the generalizability of prediction results. LOOCV allows predictive models to remain robust to the entry of new training data by numerously incorporating all observations into model estimation and thus prevents overfitting. Second, LOOCV aids in searching for the optimal combinations of hyperparameters with which ML algorithms achieve the minimum misclassification error. The identified optimal hyperparameters are to be re-used for model validation. The optimal ML hyperparameters are listed in Appendix 2.

Synthetic Minority Oversampling Technique

Categories are not equally represented in imbalanced data based on which classification is prone to be biased in favor of the majority class (Chawla et al., 2002). It is mainly because ML algorithms are trained more frequently on the majority instances and therefore tend to exhibit bias towards the majority class (Blagus & Lusa, 2013). Here, problems may arise if our primary target of prediction is the minority class. In such a case, synthetic minority oversampling technique (SMOTE) can be a useful solution. SMOTE generates newly synthesized observations based on the existing minority instances (Seo & Kim, 2018). In tackling class imbalance, some researchers may consider using random sampling methods. Random oversampling that duplicates the minority data points may pose a risk of overfitting as ML algorithms would be trained with the same observations repeatedly. Random undersampling that decreases the number of majority instances may not be a good option to consider in case of small sample size because this may lead to the significant loss of available data. Unlike the random sampling methods, SMOTE does not duplicate existing observations nor reduce the sample size. SMOTE resolves class imbalance as

well as increases sample size, leveraging the existing minority instances to synthesize new data points for the minority class. The summary below describes how SMOTE works for minority oversampling.

Synthetic minority oversampling technique (SMOTE)
1. Select a minority class instance x' in the training data at random
2. Find its k-nearest minority class neighbors
3. Choose one of the k-nearest neighbors x^k at random
4. Generate a new instance $x_{new} = (x' - x^k) * \textit{random number between 0 and 1}$
5. Repeat step 1-4 until the numbers of instances for both classes become equal

There are two SMOTE parameters – class size (m) and the number of nearest neighbors (k). m determines the number of observations in each class, and k is referenced to synthesizing new instances of each class. The original SMOTE paper (Chawla et al., 2002) suggests that the sample size of the minority class can be determined by the ratio of the number of samples in the minority class over the number of samples in the majority class. This study sets the sample sizes of both classes to be always equal (i.e., ratio = 100%). m can be set to a value greater than the sample size of the majority class, which results in oversampling of both classes. Even in this case, SMOTE synthesizes more minority instances than majority ones so that both classes are equally represented after resampling. The attempted values for two SMOTE parameters are $m = \{31, 50, 100, 300, 500, 1000\}$ and $k = \{1, 2, 3, 4, 5, 6, 7\}$. SMOTE was implemented between feature selection and predictive modeling, as attaining important features based on the original sample was preferred rather than based on partially simulated data. The *smote* function embedded in the R library *DMwR* was used in minority oversampling (Torgo, 2013).

Evaluation Metrics

Confusion Matrix

A confusion matrix presents the counts of + or – predicted classes and + or – actual classes, based on which multiple evaluation metrics are computed (Görtler et al., 2022) (Table 7). This study measures five evaluation metrics based on confusion matrix – sensitivity (SNT), specificity (SPC), precision (PRC), F-beta score (F-β), and accuracy (ACC). Considering that this project aimed at identifying those at high risk of suicide attempt, the high-risk group was labeled as the positive class (+), while the low-risk group being labeled as the negative class (–). The categories were determined with the decision threshold of .5 based on the predicted probabilities of a patient's suicidality. This means that if a predicted probability of a patient's suicidality is greater than .5, that patient is categorized as a suicide attempter (SA), otherwise, a non-attempter (NSA).

Table 7

Confusion Matrix

	Predicted classes	
Actual classes	–	+
–	true negative (TN)	false positive (FP)
+	false negative (FN)	true positive (TP)

Sensitivity (SNT), also known as recall, is the true positive rate (TPR), $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$. SNT refers to the proportion of correctly predicted positives to actual positives. Specificity (SPC) is the true negative rate (TNR), $\frac{\mathrm{TN}}{\mathrm{TN+FP}}$, referring to the proportion of correctly predicted negatives to actual

negatives. Precision (PRC) is the positive predictive value (PPV), $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$, indicating the proportion of correctly predicted positives over all predicted positives. F-β score refers to $(1+\beta^2) \times \frac{\mathrm{precision \cdot recall}}{\beta^2 \cdot \mathrm{precision+recall}}$, the weighted harmonic mean of PRC and recall (i.e., SNT). β is determined following the perception of researchers as to the relative importance of SNT to PRC. If we believe that recall should be more heavily weighed than precision, β is set to be > 1, otherwise < 1. Accuracy (ACC) refers to the number of all correct predictions over the total number of observations, $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$, which can be reformulated as 1-misclassification error.

In the present study, SNT indicates the rate at which BD patients with history of suicide attempt are correctly predicted to be suicide attempters. 1-SNT is equivalent to the type II error rate referring to the probability that a classifier incorrectly identifies high-risk patients as low-risk patients. A high type II error rate may suggest that we are at great risk of overlooking potential suicide attempters and thus missing the chance to offer timely intervention to save their lives. Meanwhile, SPC indicates the rate at which BD patients with no history of suicide attempt are correctly predicted as non-attempters. 1-SPC refers to the type I error rate indicating the probability that a classifier incorrectly identifies low-risk patients as high-risk patients. A high type I error rate would lead to overtreatment as a preventative measure, which gives rise to a far less life-threatening consequence than the cost that type II error incurs. In this sense, the acquisition of a highly sensitive classifier would be much preferred in suicide research. Furthermore, a classifier with high PRC is also sought-after because we may want positive predictions to be correctly made. Between SNT and PRC, this study prioritizes high SNT over high PRC, believing that successful retrieval of positive cases out of real positive cases is of greater importance than accurate prediction of positive cases out of positive predicted cases in suicide research. In fact, low PRC

albeit high SNT might be due to high false positives (i.e., the partial denominator of PRC). False positives might be simply attributed to the fact that those predicted to be suicide attempters have yet attempted suicide despite their high suicidal risk at the time of prognosis. Sometime in the near future, it could be just a matter of time to observe their status changes from false positives to true positives following their suicide attempt. To grant a higher weight on SNT than on PRC, β was set to 2 for F-β score. ACC is the most commonly used evaluation metric but is not the best measure in the presence of class imbalance. For example, if the size of class A far exceeds that of class B, ML algorithms will still mark high ACC even though classifiers predict most cases to class A.

Receiver Operating Characteristic curve

The receiver operating characteristic curve (ROC curve) is a two-dimensional plot that presents a balanced summary of TPR (i.e., SNT) and FPR (i.e., 1-SPC) (Cook, 2007). The ROC curve assigns observations into one of binary classes, using manifold thresholds computed based on the predicted probabilities for the positive class. For example, given ten established thresholds, ten coordinates with respect to (TPR, FPR) are projected onto the two-dimensional plane and be connected, giving rise to a ROC curve. A better binary classifier will plot the curve closer to the top left of the plane, generating a greater area under the curve (AUC). AUC measures the degree of a classifier ranking positive cases higher than negative cases at each ROC threshold (Jijkoun & Hofmann, 2009). One of the ROC thresholds serves as the optimal cut-off point that maximizes the difference between TPR and FPR (TPR $-$ FPR). In computing evaluation metrics, this study does not reference the optimal threshold but a unitary threshold, .5 across model evaluation. In real practice, a predictive model will be kept updated as more training data become available. Choosing

an optimal threshold at every update will produce varied classification results on the same observations, hampering consistent predictions across updates[2]. Therefore, this study leverages ROC curve only to attain AUC but uses .5 as the global decision threshold in calculating the other evaluation metrics.

[2] Of course, one's category can switch to the other as prediction results are updated upon the entry of new input data in spite of using the unitary cutoff point. In using a single global threshold, we only intend to prevent a change in one's predicted category simply due to a modified ROC optimal threshold as a classifier being updated.

CHAPTER 4: RESULTS

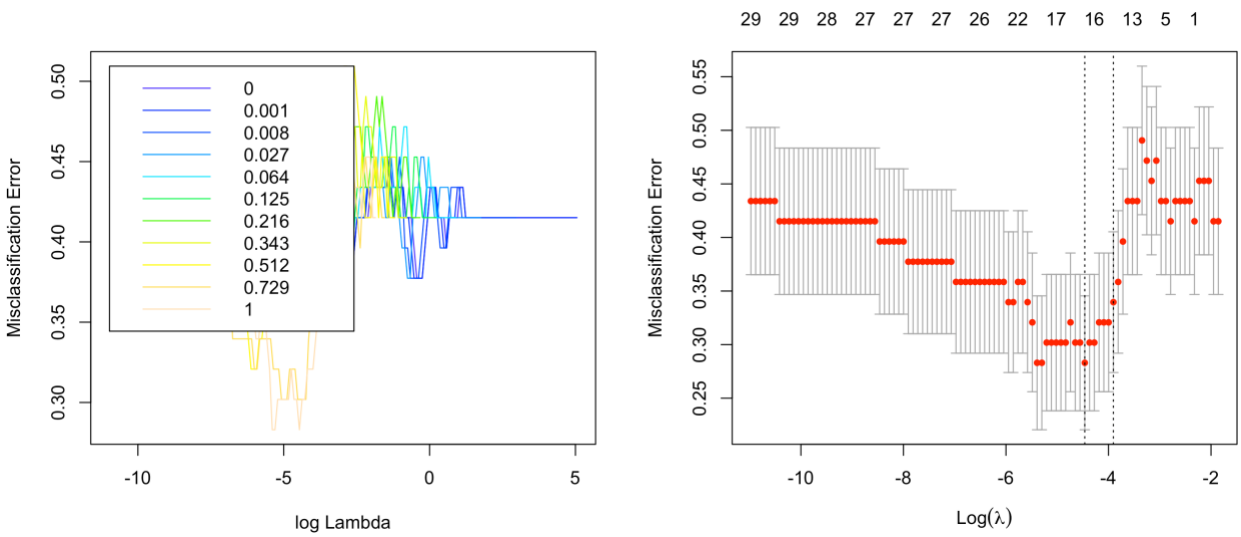
Feature Selection

What we aim to achieve by feature selection was the reduced subset of representative predictors that gives rise to the minimum misclassification error. The cross-validated ENR model showed the minimum misclassification error with sixteen selected features at α = 1 (i.e., lasso) and λ.min = .0115 (Figure 4).[3)] By the virtue of parsimony, λ.1se = .0202 that refers to the largest λ at which the misclassification error rate lies within one standard error away from the minimum misclassification error was chosen over λ.min (Figure 4). Therefore, at α = 1 and λ.1se = .0202, fifteen ERPs were selected as the final feature set that enters the predictive modeling process going forward (Table 8). It is notable that the majority of the selected ERPs were extracted from midline and right side of the brain according to laterality and the central and parietal brain region according to caudality, but no features originated from the left frontal region. The Bonferroni corrected t-tests were conducted on each feature to further examine significant differences in the selected ERP latencies between SA and NSA, but no significant mean differences were discovered. Such statistical non-significance may be attributed to the low power due to the small sample size. As aforementioned, ENR is an embedded method that simultaneously performs predictive modeling in the course of feature selection. The prediction performance of ENR is summarized in the results of analysis 1 (Table 9).

[3] In a data-driven manner, ENR performed the best when lasso was chosen by cross-validation, but this is not always the case. Applying it to other empirical data may give rise to a different value of α rather than 1.

Figure 4

Tuning hyperparameters of ENR and feature selection



Note. (Left) ENR. The *cva.glmne*t function in the R *glmnet* library simultaneously cross-validates, seeking for the optimal values of α and λ in ENR. The misclassification error was minimized by lasso ($\alpha = 1, \lambda.min = .0115$). (Right) Lasso. The plot on the right-hand side provides more details about the LASSO result. The upper x-axis indicates the number of selected features, and the lower x-axis shows the range of log(λ). The left dotted vertical line shows the pairing of λ.min and the number of selected features at the corresponding λ.min. The right dotted vertical line indicates the pairing of λ.1se ($\lambda = .0202$) and the number of selected variables at λ.1se. In this study, λ.1se was chosen over λ.min by the virtue of parsimony. In the end, 15 ERPs were chosen as the final feature set for predictive modeling.

Table 8

Fifteen ERPs selected by ENR

Laterality / Caudality	3 (Left)	z (Midline)	4 (Right)
F (front)	-	FzS1, FzS4	F4S2
C (central)	C3S1	CzS4	C4S1, C4S3, C4S4
P (parietal)	P3S1, P3S2	PzS1, PzS3, PzS4	P4S1, P4S2

Predictive Modeling

Analysis 1. Predicting Suicide Attempt with Original N200 ERPs

Analysis 1 examines the efficacy of feature selection prior to predictive modeling, comparing the prediction performances of ML models on the original 36 ERPs (D1) to those on the selected features of 15 ERPs (D2). The results showed that feature selection by ENR noticeably improved the prediction performance of the trained ML algorithms across the board (Figure 5). To begin with, considerable increase in most of the evaluation metrics was observed from the predictive models with D2. In particular, the severe imbalance between SNT and SPC seen in the outcomes with D1 was largely resolved with D2. That is, predictive models constructed with D2 showed superior performance in identifying patients with high suicidality, whereas the D1-based models performed better in detecting patients with low suicidality. Also, feature selection by ENR contributed not only to increase the magnitudes of evaluation metrics but also to reduce their variability across the algorithms. Meanwhile, ENR demonstrated poor performance in prediction, merely achieving 66.7% AUC, 50.0% SNT, and 61.1% PRC. Those empirical results support the current framework taking two separate phases for feature selection with embedded methods (e.g.,

ENR or lasso) and predictive modeling with other supervised learning algorithms, rather than solely counting on the embedded methods for both purposes.

Figure 5

Comparison of prediction performance on D1 and D2

Note. Each box plot represents the value of the corresponding metrics that six main ML algorithms achieved via predictive modeling with D1 and D2. D1 represents the original 36 ERPs. D2 indicates the 15 ERPs selected by ENR.

Now we turn to the detailed prediction outcomes of six main ML algorithms with D2 (Table 9). DNN algorithm produced the top performing model achieving 84.5% AUC, 81.8% SNT, 72.0% PRC, 79.6% F-β score, 77.4% SPC, and 79.2% ACC. This result is promising given that even with the small sample, the DNN model achieved high SNT and AUC over 80% and satisfactory PRC over 70%. In addition, SNT recorded even higher than SPC despite the presence of slight imbalance biased towards the negative class (i.e., non-attempter). The XGBoost model also achieved high AUC over 80%, but SPC (77.4%) recorded slightly higher than SNT (72.7%). The rest of the models produced even more biased outcomes towards NSA. For example, the radial SVM model merely achieved 54.5% SNT despite high AUC and SPC over 80%. Such biased outcomes are likely due to the skewed class proportions towards the negative class even if the imbalance was not severe.

The results of analysis 1 can be encapsulated as follows. First, feature selection before predictive modeling gave rise to a smaller subset of predictors that considerably improved prediction performance. Second, the two-step approach that employs feature selection prior to predictive modeling demonstrated its superior effectiveness in acquiring competent predictive models to all-in-one approach that solely counts on a regularization algorithm (e.g., ENR or lasso) for both feature selection and predictive modeling. Third, the DNN model outperformed the other ML models, achieving the highest SNT and AUC. Fourth, the non-DNN models achieved higher SPC than SNT, which is not sought-after in suicide research that mainly aims to identify potential suicide attempters. Such a biased prediction outcome might be due to a greater number of negative cases in this small sample. In an attempt to tackle this, we go ahead to analysis 2 and add the minority oversampling phase between feature selection and predictive modeling.

Table 9

Summary of analysis 1

ML algorithms	XGBoost		DNN		Linear SVM		Radial SVM		Poly SVM		LR		ENR	
Dataset	**D1**	**D2**	**D1**	**D2**	**D1**	**D2**	**D1**	**D2**	**D1**	**D2**	**D1**	**D2**	**D1**	**D2**
AUC	61.9	83.4	66.1	84.5	29.5	80.4	49.4	83.1	47.5	76.1	44.9	71.0	66.7	-
SNT	50.0	72.7	54.5	81.8	4.5	54.6	18.2	54.5	13.6	50.0	40.9	72.7	50.0	-
PRC	52.4	69.6	60.0	72.0	33.3	75.0	50.0	66.7	50.0	68.8	36.0	72.7	61.1	-
F-beta	50.5	72.1	55.6	79.6	5.5	57.7	20.8	56.6	16.0	52.9	39.8	72.7	51.9	
SPC	67.7	77.4	74.2	77.4	93.5	87.1	87.1	80.6	90.3	83.9	48.4	80.6	77.4	-
ACC	60.4	75.5	66.0	79.2	56.6	73.6	58.5	69.8	58.5	69.8	45.3	77.4	66.0	-

Note. D1 indicates the original dataset of 36 ERPs. D2 is the subset of 15 ERPs selected by ENR. F-beta score was calculated with $\beta = 2$. The D1 column of ENR presents the prediction outcomes that came along during feature selection. Given that ENR was used only for the purpose of feature selection in this study, ENR was not run for further analyses with D2-D4.

Analysis 2. Predicting Suicide Attempt with Oversampled N200 ERPs

Analysis 2 implements the integrative ML approach that the current study ultimately attempts to test and explores the efficacy of oversampling by SMOTE as a solution to further improve SNT in the predictive models. From the prior analysis, biased prediction results towards the NSA class were observed across the majority of the ML models. It was assumed that severe imbalance between SNT and SPC might be attributed to the slight class imbalance between NSA and SA. SMOTE can offer a solution to imbalanced data by generating new data points for the minority class. Multiple values for two SMOTE parameters – class size, $m = \{31, 50, 100, 300, 500, 1000\}$, and the number of nearest neighbors, $k = \{1, 2, 3, 4, 5, 6, 7\}$, were tried in minority oversampling. Here the oversampled D2 is referred to as D3. The optimal values of hyperparameters attained with D2 were re-used for predictive modeling based on D3 in order to make the analytical process less computationally intensive and to focus on probing for the optimal pairs of SMOTE parameters in this analysis.

Prediction with D3 demonstrated noticeable improvement in most of the evaluation metrics, when compared to the outcomes with D2 (Table 10) (Figure 6). First, SNT largely improved across the algorithms: linear SVM (54.6% → 77.3%), radial SVM (54.5% → 86.4%), polynomial SVM (50.0% → 81.8%), XGBoost (72.7% → 86.4%), LR (72.7% → 77.3%). DNN, the best predictive model previously with D2, maintained the same level of SNT at 81.8%. Second, the variability of the evaluation metrics was considerably reduced after SMOTE, meaning that it has gotten harder to rank the classifiers according to their performance. The large increase in SNT pushed upwards ACC and F-β score altogether. PRC remained almost unchanged even though achieving high precision has gotten more challenging with the increased numbers of cases predicted SA. The overall levels of SPC slightly reduced after oversampling. But it may not necessarily mean that the

classifiers built on D3 predicted NSA less accurately. Rather, SPC decreased as the candidate models with high SPC did not pass the model selection criteria described below.

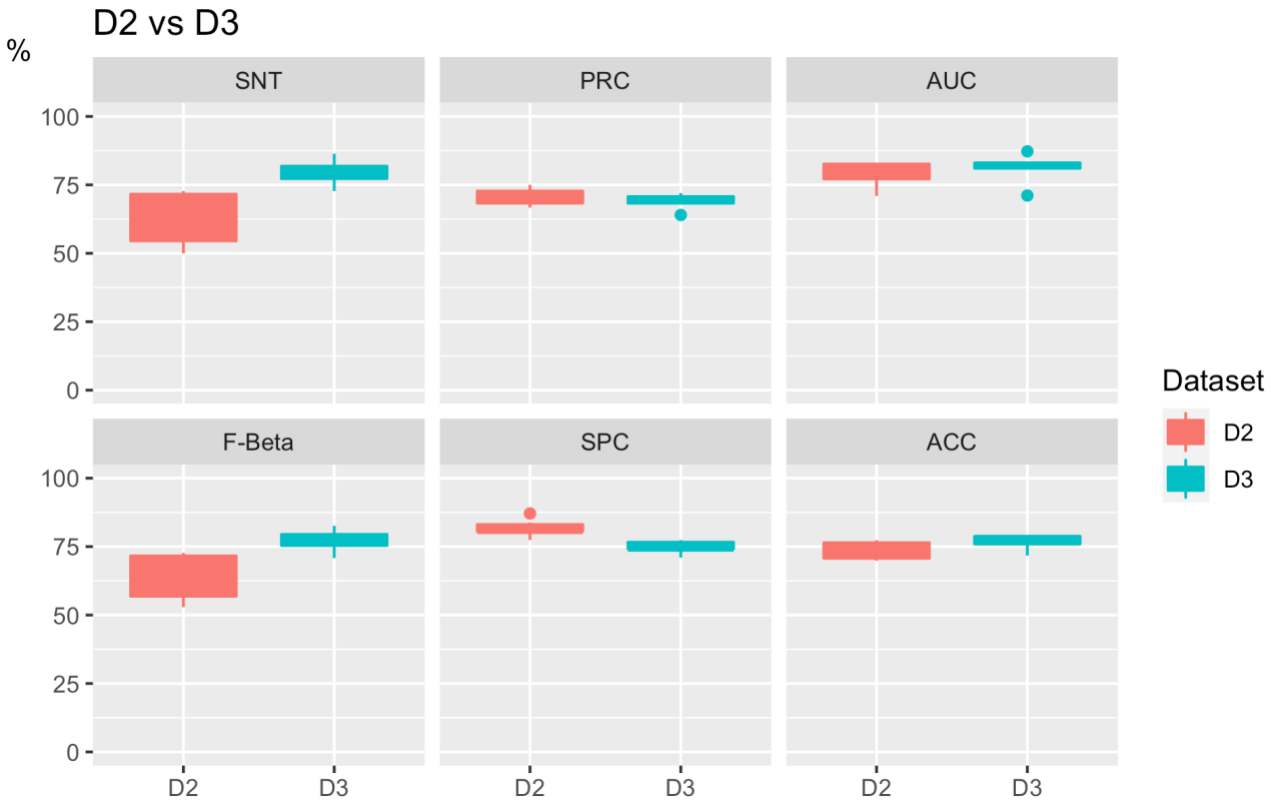
Out of 42 candidate models by algorithm, the representative models were selected, sequentially passing the model selection criteria – 1) top 10 models according to SNT, 2) the top 5 models with respect to PRC out of the selection passing criterion 1, 3) top 1 model with respect to AUC out of those retained by criterion 2. We will want a predictive model that accurately identifies potential suicide attempters but not overly produce many false positive cases. For this, the model should achieve high SNT, PRC, and AUC. The order of the selection criteria reflects our differential preference on the evaluation metrics such as SNT > PRC > AUC. The prediction results by the representative models that passed the selection criteria are listed in Table 10. The best model of the representatives was the radial kernel SVM that achieved 86.4% SNT, 70.4% PRC, 87.2% AUC, 82.6% F-β score, 74.2% SPC, and 79.3% ACC. The XGBoost model ranked the second-best with 86.4% SNT, 70.4% PRC, and 83.4% AUC. The other classifiers also achieved high AUC and SNT over 70% with SNT being equivalent to or higher than SPC. Meanwhile, there seemed to be no omnipotent combination of SMOTE hyperparameters that maximized the performance of all the algorithms. That is, each algorithm showed the most idealistic performance with different combinations of (k, m). The optimal values of k varied across the models, but no representative model was achieved given m = 500 or 1,000. This may imply that excessive increase in sample size perhaps led the oversampled data to be far deviated from D2, so applying the hyperparameters tuned on D2 no longer gave rise to the best predictive models with D3.

Analysis 2 can be summarized as follows. First, generating new data points for the minority class by SMOTE further pushed SNT upwards across the board. Second, the variability of the evaluation metrics mostly decreased. Third, the radial kernel SVM model was the top performing

model with the highest SNT (86.4%) and AUC (87.2%), and satisfactory PRC (70.4%). Fourth, the logistic regression model was the worst performing model but still demonstrated sound performance with the metrics being nearly or above 70%. Finally, there was no general consensus in regard to the optimal values of SMOTE parameters, but an overly increase in the sample size did not seem to further enhance prediction.

Figure 6

Comparison of prediction performance on D2 and D3



Note. Box plots represent evaluation metrics across six ML algorithms given D2 and D3. D2 is the reduced subset of ERPs by feature selection. D3 is the oversampled D2.

Table 10

Summary of analysis 2

ML algorithms	XGBoost		DNN		Linear SVM		Radial SVM		Poly SVM		LR	
Dataset	**D2**	**D3**	**D2**	**D3**	**D2**	**D3**	**D2**	**D3**	**D2**	**D3**	**D2**	**D3**
AUC	83.4	83.4	84.5	82.3	80.4	80.6	83.1	87.2	76.1	82.7	71.0	71.1
SNT	72.7	86.4	81.8	81.8	54.6	77.3	54.5	86.4	50.0	81.8	72.7	77.3
PRC	69.6	70.4	72.0	66.7	75.0	70.8	66.7	70.4	68.8	69.2	72.7	68.0
F-beta	72.1	82.6	79.6	78.3	57.7	75.9	56.6	82.6	52.9	79.0	72.7	75.2
SPC	77.4	74.2	77.4	71.0	87.1	77.4	80.6	74.2	83.9	74.2	80.6	74.2
ACC	75.5	79.3	79.2	75.5	73.6	77.4	69.8	79.3	69.8	77.4	77.4	75.5
$\boldsymbol{m}$	-	100	-	31	-	50	-	31	-	50	-	300
$\boldsymbol{k}$	-	6	-	7	-	5	-	1	-	3	-	1

Note. D2 consists of 15 ERPs selected by ENR. D3 denotes the oversampled D2 by SMOTE. m and k are SMOTE parameters. m indicates sample size for each class. k is the number of nearest neighbors.

Analysis 3. Do ERPs Predict Suicide Attempt Better than Demographics?

Analysis 3 investigates whether ERPs serve as the superior predictors of suicide attempt to demographic variables, which have long served as traditional predictors in suicide research to date. As to the prediction outcomes of ERPs, the results from D2 were borrowed. And six predictive models were generated, using D4 that consists of three demographic variables – gender, age, and years of education. In that most of the six ML algorithms do not accept categorical variable as predictors, the gender variable was pre-processed to be numerical, -1 (male) or 1(female). In addition, there were two missing values in years of education. As a result, demographic variables from 51 participants entered predictive modeling process.

Table 11 and Figure 7 present the prediction outcomes with D2 and D4. Only the DNN model built on D4 showed the performance slightly better than random guessing, merely hitting the 60% level. The other ML models demonstrated almost null capability of detecting SA cases given too low SNT less than 20%. On the contrary, SPC recorded very high, indicating that most of the ML models based on D4 pushed most cases into the NSA class. This analysis suggests that demographic variables alone would not bring fruitful prediction outcomes in identifying potential suicide attempters, and ERPs should be further investigated in larger samples so that we can assure its promise as an effective neural marker of suicide risk.

Table 11

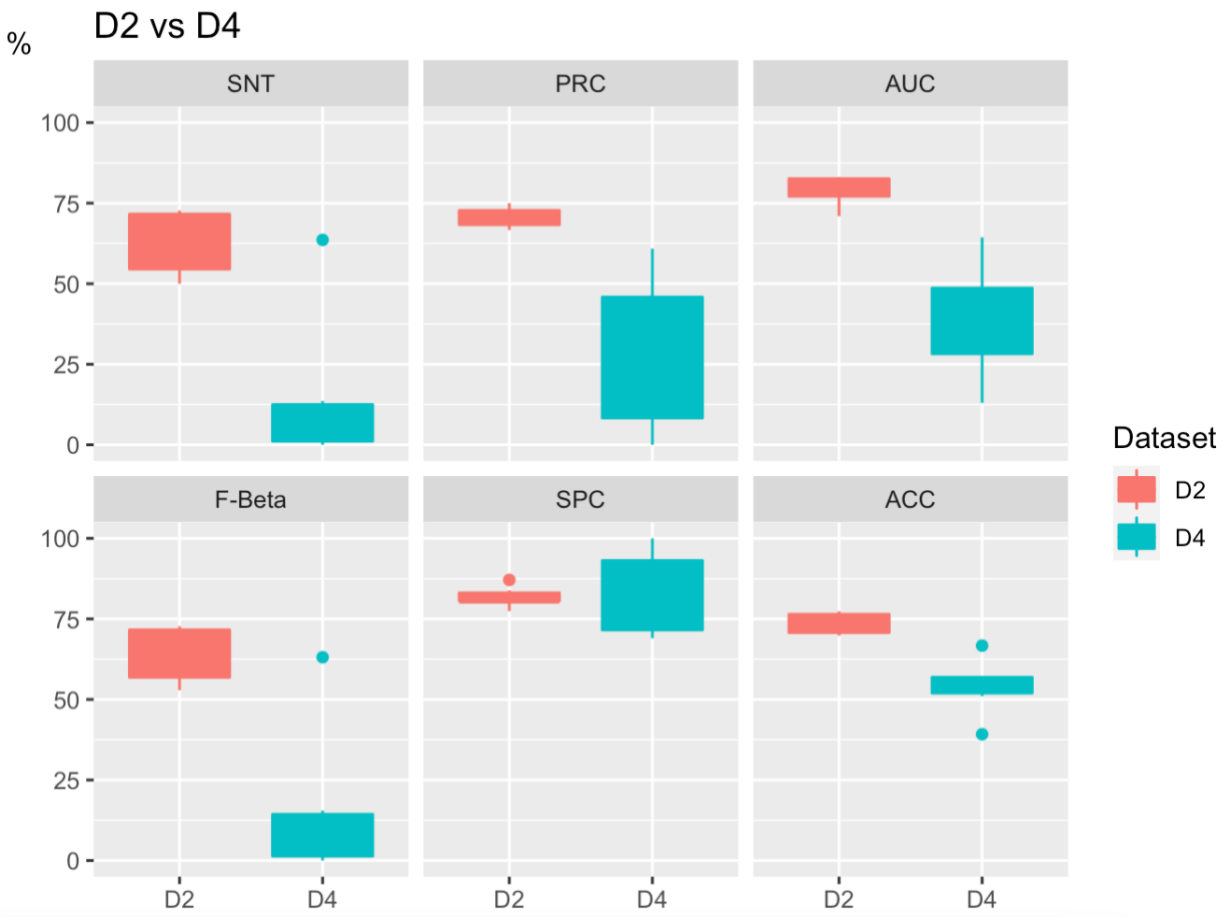
Summary of analysis 3

ML algorithms	XGBoost		DNN		Linear SVM		Radial SVM		Poly SVM		LR	
Dataset	**D2**	**D4**	**D2**	**D4**	**D2**	**D4**	**D2**	**D4**	**D2**	**D4**	**D2**	**D4**
AUC	82.8	13.0	84.5	64.4	80.4	43.1	83.1	50.5	74.8	27.1	71.0	31.7
SNT	72.7	0	81.8	63.6	54.6	13.6	54.5	9.1	50.0	4.5	72.7	0
PRC	72.7	0	72.0	60.9	75.0	33.3	66.7	50.0	68.8	33.3	72.7	0
F-beta	72.7	0	79.6	63.1	57.7	15.5	56.6	10.9	52.9	5.5	72.7	0
SPC	80.6	100.0	77.4	69.0	87.1	79.3	80.6	93.1	83.9	93.1	80.6	69.0
ACC	77.4	56.8	79.2	66.7	73.6	51.0	69.8	56.9	69.8	54.9	77.4	39.2

Note. D2 indicates the reduced set of predictors by feature selection. D4 consists of demographic variables including age, gender, and years of education.

Figure 7

Comparison of prediction performance on D2 and D4



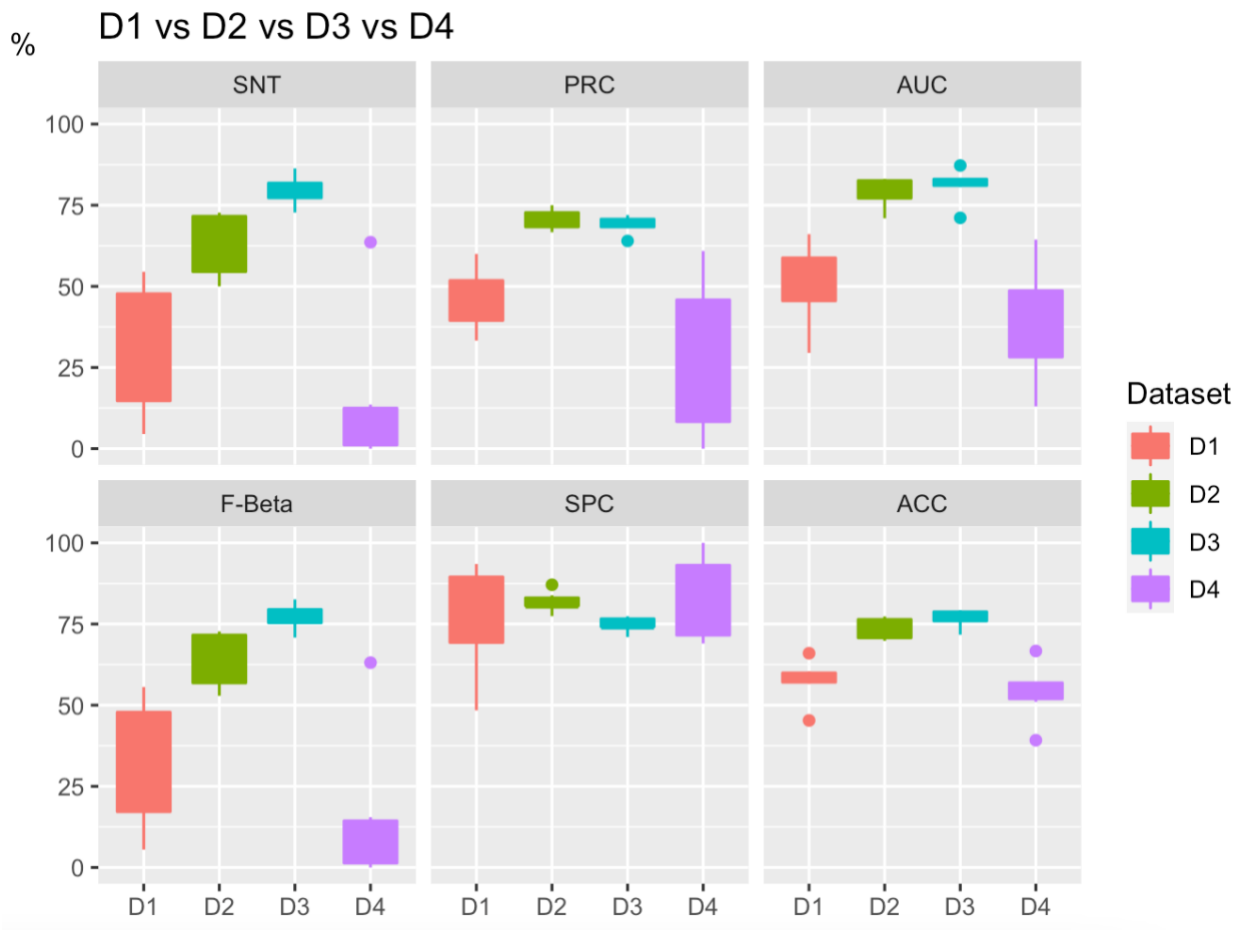
Note. The plot represents the prediction performance of six ML algorithms given D2 and D4. D2 consists of fifteen ERPs selected by ENR. D4 contains three demographic variables.

Conclusion

Across three analyses, supervised ML algorithms generated classifiers based on four different sets of predictors - 36 original ERPs (D1), 15 ERPs selected by ENR (D2), oversampled D2 by SMOTE (D3), and demographic variables (D4). The comprehensive plots on the prediction results with D1-D4 are shown in Figure 8 and Appendix 3. By and large, prediction on D3 demonstrated the best results with the highest evaluation metrics and the lowest variability across the performances of ML algorithms. This may suggest that consolidation modeling that integrates feature selection, minority oversampling, and supervised ML can offer a reliable classifier that effectively predicts person-specific suicidality when we are in need of properly addressing small samples and high-dimensional imbalanced data. Furthermore, the contrasting prediction results with ERP data (D1-D3) and demographic data (D4) imply that ERPs might serve as a promising neural marker for a person's suicidal risk.

Figure 8

Summary of overall prediction results



Note. The figure represents the averaged performance of the ML predictive models built with D1 to D4. Prediction with D3 brought us the most promising results, in general, implied by evenly high evaluation metrics and their low variability. Predictive models with D4 showed almost null results except SPC. This summary plot well demonstrates the effectiveness of the integrative ML approach for classification in tackling small samples with high dimensional imbalanced data and the latent power of ERPs as a tracer of suicidality.

CHAPTER 5: DISCUSSION

The overarching goal of this study was to put forward an integrative machine learning approach – feature selection, minority oversampling, and predictive modeling – devised to generate powerful predictive models (i.e., classifiers) for a small dataset with high-dimensionality and class imbalance that psychologists often encounter in conducting research. We can summarize the current study with five main conclusions. First, the proposed approach enabled us to build promising predictive models via deep neural network, extreme gradient boosting, and radial kernel support vector machine, achieving high AUC and sensitivity over 80%. Second, it is noteworthy that feature selection before predictive modeling improved not only the prediction performance of classifiers but also their interpretability, giving information on which features were importantly used in predictive modeling. Third, minority oversampling on the selected features gave rise to the best predictive models in general with the machine learning algorithms in use. Fourth, leave-one-out cross-validation helped exploit the small sample to the maximum so as to achieve better generalizable models. Fifth, event-related potentials produced more fruitful prediction outcomes than demographic variables.

As demonstrated in this paper, prediction by machine learning can still be viable even for small samples and high-dimensional imbalanced data by the analytical processes described herein. One thing to note is that the choice of important evaluation metrics would vary, depending on the interest of research. Sensitivity took precedence over the other indicators in this study because its

primary goal was the correct identification of potential suicide attempters. Different research may necessitate different priority such as precision in preference to sensitivity.

Further extensions can be made to overcome the limitations of the current study. To begin with, multiple machine learning algorithms employed in this study were the ones that had been at the forefront at the time of analyses. Since there will be always new ones offered in the future, an even wider range of machine learning algorithms should be explored in an attempt to secure more powerful predictive models. Second, it is still recommended to widen the pool of samples to be freed from any issues related to selection bias. The sample in use was collected from bipolar disorder patients participating in a specific research program, so they might not serve as the representative of the target population. Third, suicide attempters may show different time intervals between attempted suicide and ERP measurement, but such discrepancy was not factored into prediction. If the exact time when suicide was attempted gets available, controlling the interval between the ERP measurement and attempted suicide would allow us to test the efficacy of predictors more thoroughly. Finally, forecasting the expected time of the target behavior based on repeated measures of predictors would constitute another crucial axis of prediction research. Forecasting would help complement classification approach that frequently gives rise to falsely predicted cases. For example, suicide attempt largely predating ERP measurement might have led to false negative cases, since the recent ERPs may not properly reflect one's suicidality that once used to be strong but not anymore. In addition, false positive cases might have arisen from suicide attempt lagging ERP measurement. In this case, the transition from false positive to true positive could be just a matter of time. Therefore, forecasting based on time series data would propel prediction research further, providing the useful means to capture any critical temporal patterns that may be seen when a target behavior is imminent.

APPENDIX 1: TABLE OF DNN ACTIVATION FUNCTIONS

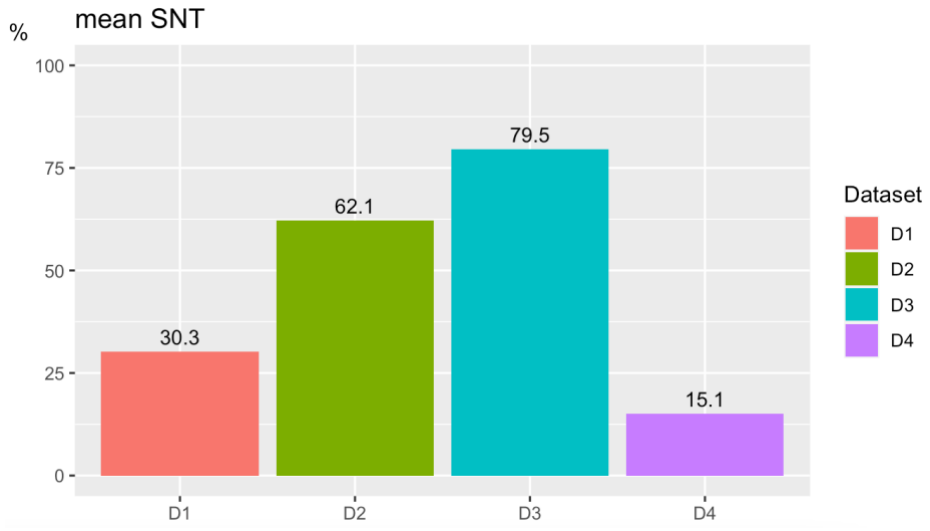
Activation function	$\boldsymbol{f(x)}$	**Properties**
Binary step	$f(x) = 1, if\ x \geq 0$ $f(x) = 0, otherwise$	simplest activation function not feasible in multiclass classification backpropagation unavailable
Binary	$f(x) = ax\ (a\ is\ constant)$	only linear classification cannot identify complex patterns from data
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	non-linear function for binary classification not symmetric about zero
Tanh	$f(x) = 2\left(\frac{1}{1+e^{-2x}}\right) - 1$	similar with Sigmoid function symmetric about zero
ReLU	$f(x) = \max(0, x)$	rectified linear unit non-linear function efficient computation due to sparsity hidden layer only
Leaky ReLU	$f(x) = 0.01x, \quad if\ x < 0$ $f(x) = x, \quad otherwise$	variant of ReLU function non-zero value for negative values of x
ELU	$f(x) = x, if\ x \geq 0$ $f(x) = a(ex - 1), otherwise$	exponential linear unit variant of ReLU function introduction of a slope for the negative values of x
Softmax	$f(\boldsymbol{x})_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}$ $(j = 1, \ldots, K)$	An extended version of Sigmoid function often used in the output layer

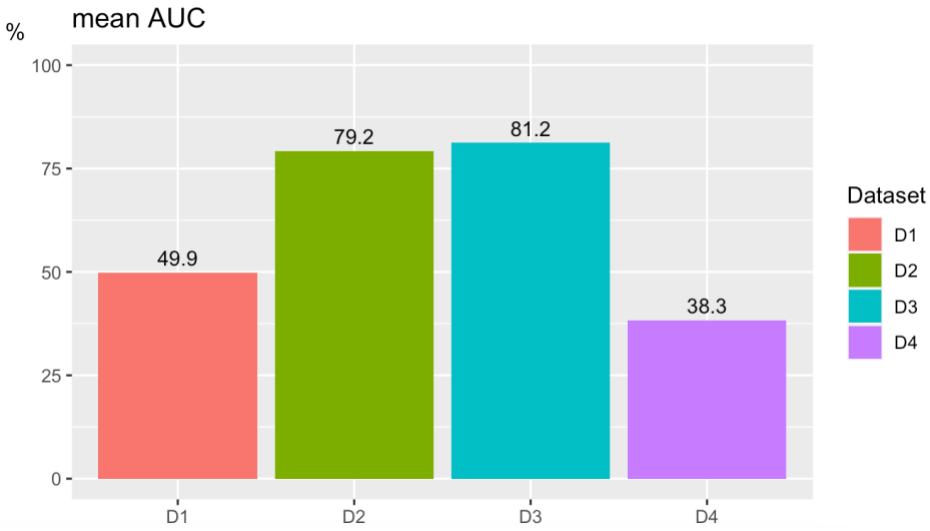
Note. Nwankpa et al., 2018; Sharma et al., 2017

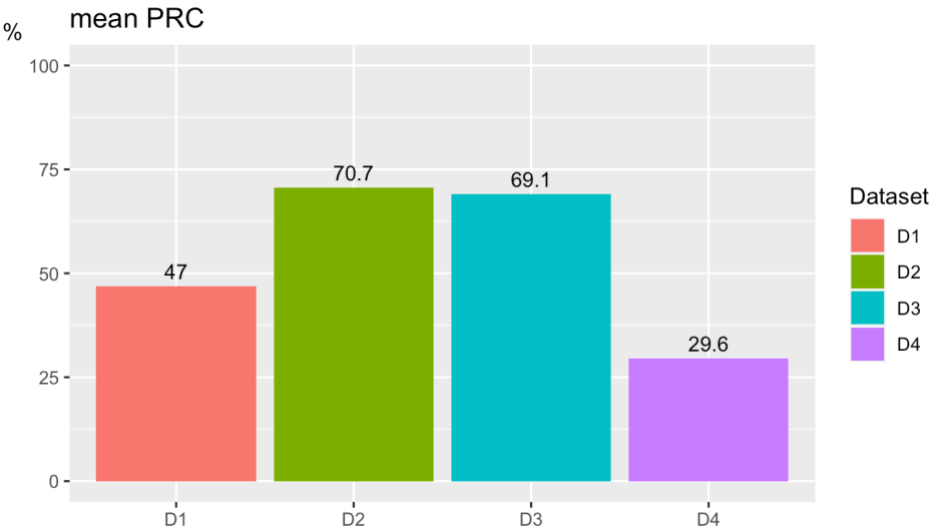
APPENDIX 2: TABLE OF OPTIMAL HYPERPARAMETERS

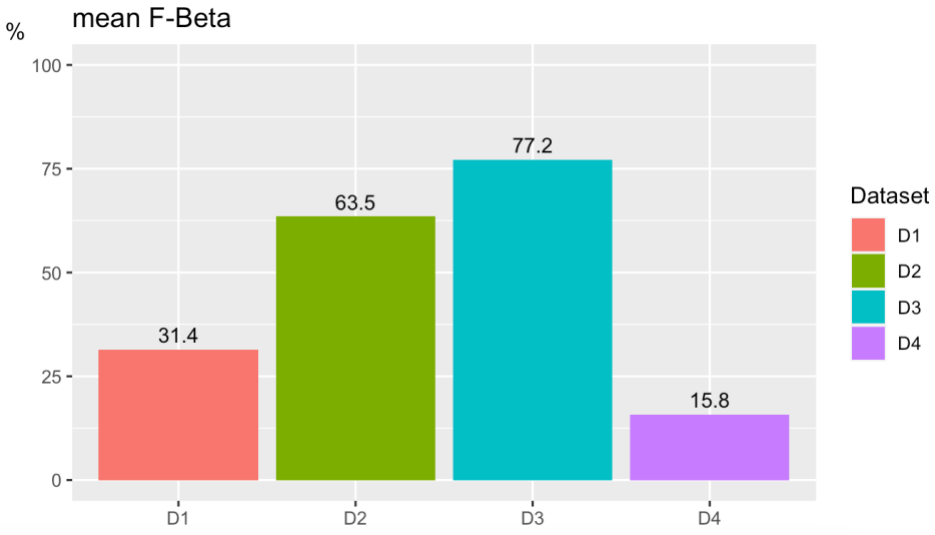
Algorithms	Parameters	D1	D2	D3	D4
ENR	**alpha**	-	1	1	1
	gamma	-	.0202	.0202	.0202
Linear SVM	**cost**	1	5	5	.1
Radial SVM	**cost**	1,000	10	10	1,000
	gamma	.0001	.01	.01	.1
Polynomial SVM	**cost**	.01	100,000	100,000	100,000
	gamma	3	.0005	.0005	.1
	degree	1	1	1	5
DNN	**number of hidden layers**	2			
	units	input = 128; hidden 1 = 64; hidden 2 = 32; output = 2			
	activation	hidden = ReLU; output = softmax			
	optimizer	adam			
	loss function	categorical cross entropy			
	metrics	accuracy			
	epoch	300	300	300	300
	weight decay	.01	.01	.01	.01
	dropout	input = .1 hidden 1 = .3 hidden 2 = .3	input = .3 hidden 1 = .1 hidden 2 = .1	input = .3 hidden 1 = .1 hidden 2 = .1	input = .1 hidden 1 = .1 hidden 2 = .1
	learning rate	.01	.01	.01	.01
	batch size	30	30	30	30
XGBoost	**booster**	gblinear			
	objective	binary::logistic			
	eval_metric	logloss			
	nrounds	10	30	30	10
	lambda	0	0	0	.5
	gamma	0	0	0	0
LR	**no parameters**	-			

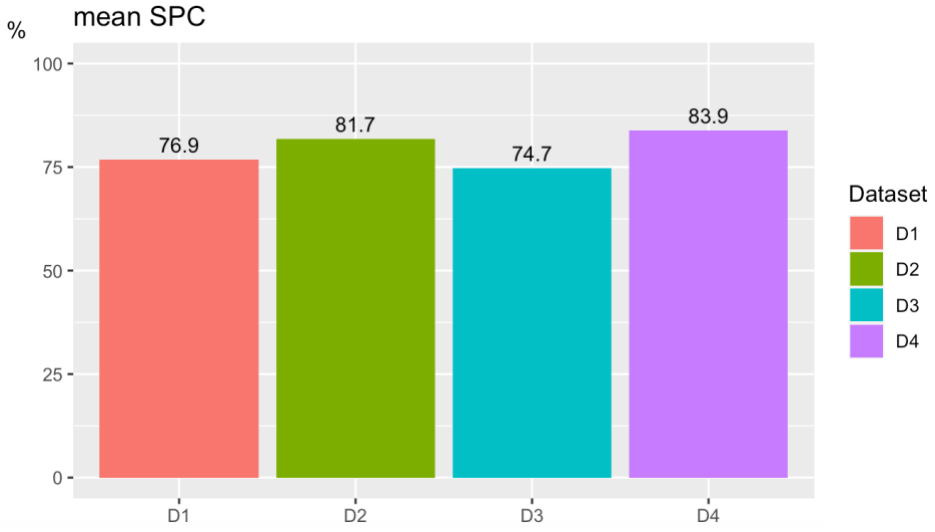
APPENDIX 3: FIGURES OF MEAN EVALUATION METRICS BY DATASET

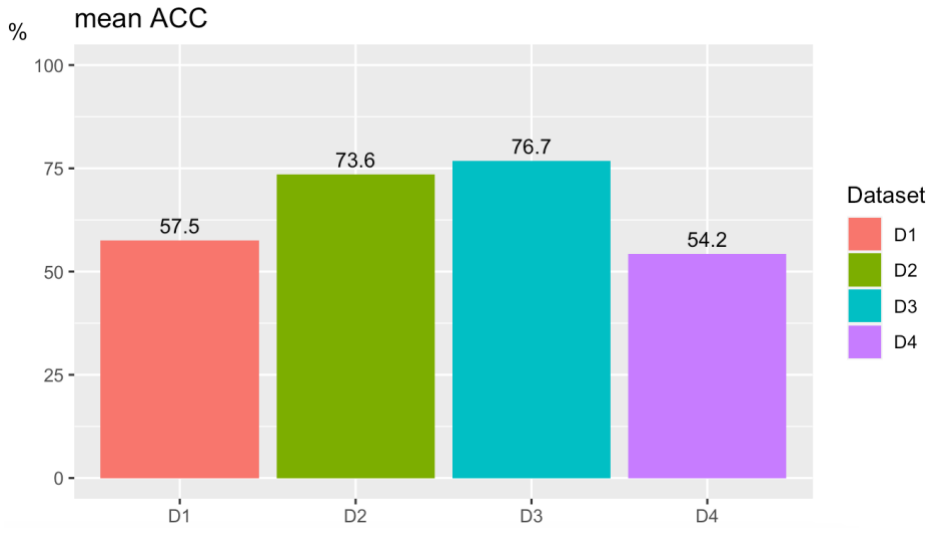







mean F-Beta
%
100
75
50
25
0
31.4
63.5
77.2
15.8
D1
D2
D3
D4
Dataset
D1
D2
D3
D4


mean SPC
%
100
75
50
25
0
76.9
81.7
74.7
83.9
D1
D2
D3
D4
Dataset
D1
D2
D3
D4


mean ACC
%
100
75
50
25
0
57.5
73.6
76.7
54.2
D1
D2
D3
D4
Dataset
D1
D2
D3
D4

REFERENCES

Albanese, B. J., Macatee, R. J., Gallyer, A. J., Stanley, I. H., Joiner, T. E., & Schmidt, N. B. (2019). Impaired conflict detection differentiates suicide attempters from ideating nonattempters:Evidence from event-related potentials. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, *4*(10), 902-912. doi: 10.1016/j.bpsc.2019.05.015

Albert, J., López-Martín, S., Tapia, M., Montoya, D., & Carretie, L. (2012). The role of the anterior cingulate cortex in emotional response inhibition. *Human Brain Mapping*, *33*(9), 2147-2160. doi:10.1002/hbm.21347

APA (2013). Diagnostic and statistical manual of mental disorder (5th ed.), *American Psychiatric Association*, 21(21), 591-643.

Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is machine learning? A primer for the epidemiologist. *American Journal of Epidemiology*, *188*(12), 2222-2239. doi: 10.1093/aje/kwz189

Bickel, P. J., Li, B., Tsybakov, A. B., van de Geer, S. A., Yu, B., Valdés, T., Rivero, C., Fan, J., & Van der Vaart, A. (2006). Regularization in statistics. *Test*, *15*(2), 271-344. doi: 10.1007/BF02607055

Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, *14*(1), 1-16. doi:10.1186/1471-2105-14-106

Bohaterewicz, B., Sobczak, A.M., Podolak, I., Wójcik, B., Mętel, D., Chrobak, A. A., Fafrowicz, M., Siwek, M., Dudek, D., & Marek, T. (2021). Machine learning-based identification of suicidal risk in patients with schizophrenia using multi-level resting-state fMRI features. *Frontiers in Neuroscience*, *14*, 605697. doi:10.3389/fnins.2020.605697

Bokura, H., Yamaguchi, S., & Kobayashi, S. (2001). Electrophysiological correlates for response inhibition in a Go/no-go task. *Clinical Neurophysiology*, *112*(12), 2224-2232. doi: 10.1016/S1388-2457(01)00691-5

Bostwick, J. M., Pabbati, C., Geske, J. R., & McKean, A. J. (2016). Suicide attempt as a risk factor for completed suicide: even more lethal than we knew. *American Journal of Psychiatry*, *173*(11), 1094-1100. doi: 10.1176/appi.ajp.2016.15070854

Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, *24*(6), 2350-2383. doi: 10.1214/aos/1032181158

Button, K. S., Ioannidis, J., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S., & Munafò, M. R. (2013). Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, *14*(5), 365-376. doi: 10.1038/nrn3475

Chapman, R. M., & McCrary, J. W. (1995). EP component identification and measurement by principal components-analysis. *Brain and Cognition*, *27*(3), 288-310. doi: 10.1006/brcg.1995.1024

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321-357. doi: 10.1613/jair.953

Chen, Q., Zhang-James, Y., Barnett, E. J., Lichtenstein, P., Jokinen, J., D'Onofrio, B. M., Faraone, S.V., Larsson, H. & Fazel, S. (2020). Predicting suicide attempt or suicide death following a visit to psychiatric specialty care: A machine learning study using Swedish national registry data. *PloS Medicine*, *17*(11), e1003416. doi:10.1371/journal.pmed.1003416

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., … & Yuan, J. (2021). Package 'xgboost', https://cran.r-project.org/web/packages/xgboost/xgboost.pdf

Christodoulou, T., Lewis, M., Ploubidis, G. B., & Frangou, S. (2006). The relationship of impulsivity to response inhibition and decision-making in remitted patients with bipolar disorder. *European Psychiatry*, *21*(4), 270-273. doi:10.1016/j.eurpsy.2006.04.006

Chun, J., Karam, Z. N., Marzinzik, F., Kamali, M., O'Donnell, L., Tso, I. F., ... & Deldin, P. J. (2013). Can P300 distinguish among schizophrenia, schizoaffective and bipolar I disorders? An ERP study of response inhibition. *Schizophrenia Research*, *151*(1-3), 175-184. doi: 10.1016/j.schres.2013.10.020

Coles, M. G., & Gratton, G. (1986). Cognitive psychophysiology and the study of states and processes. *Energetics and Human Information Processing*. NATO ASI Series, vol 31. Springer, Dordrecht. 409-424. doi: 10.1007/978-94-009-4448-0_29

Cook, N. R. (2007). Use and misuse of the receiver operating characteristic curve in risk prediction. *Circulation*, *115*(7), 928-935. doi: 10.1161/CIRCULATIONAHA.106.672402

Ding, B., Qian, H., & Zhou, J. (2018). Activation functions and their characteristics in deep neural networks. *2018 Chinese Control and Decision Conference (CCDC)*, IEEE. 1836-1841.doi: 10.1109/CCDC.2018.8407425

Dome, P., Rihmer, Z., & Gonda, X. (2019). Suicide risk in bipolar disorder: a brief review. *Medicina*, *55*(8), 403. doi: 10.3390/medicina55080403

Dominke, C., Graham-Schmidt, K., Gentsch, A., & Schütz-Bosbach, S. (2021). Action inhibition in individuals with high obsessive-compulsive trait of incompleteness: An ERP study. *Biological Psychology*, 159, 108019. doi: 10.1016/j.biopsycho.2021.108019

Donchin, E., & Coles, M. G. (1988). Is the P300 component a manifestation of context

updating?. *Behavioral and Brain Sciences*, *11*(3), 357-374. doi: 10.1017/S0140525X00058027

Eimer, M. (1993). Effects of attention and stimulus probability on ERPs in a Go/no-go task. *Biological Psychology*, *35*(2), 123-138. doi: 10.1016/0301-0511(93)90009-W

Falbel, D., Allaire, J.J., Rstudio, Tang, Y., Eddelbuettel, D., Golding, N., Kalinowski, T., Google Inc. (2021): Package 'tensorflow', https://cran.r-project.org/web/packages/tensorflow/tensorflow.pdf

Fan, P., Guo, X., Qi, X., Matharu, M., Patel, R., Sakolsky, D., ... & Wang, L. (2020). Prediction of suicide-related events by analyzing electronic medical records from PTSD patients with bipolar disorder. *Brain Sciences*, *10*(11), 784. doi: 10.3390/brainsci10110784

Franklin, J. C., Ribeiro, J. D., Fox, K. R., Bentley, K. H., Kleiman, E. M., Huang, X., Musacchio, K.M., Jaroszewski, A. C., Chang, B.P. & Nock, M. K. (2017). Risk factors for suicidal thoughts and behaviors: A meta-analysis of 50 years of research. *Psychological Bulletin*, *143*(2), 187. doi: /10.1037/bul0000084.

Freijeiro-González, L., Febrero-Bande, M., & González-Manteiga, W. (2022). A critical review of LASSO and its derivatives for variable selection under dependence among covariates. *International Statistical Review*, *90*(1), 118-145. doi: 10.1111/insr.12469

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367-378. doi: 10.1016/S0167-9473(01)00065-2

Friedman, J. H., Hastie, T., Simon, N., Tibshirani, R., Hastie, M. T., & Matrix, D. (2017). Package 'glmnet.'. *Journal of Statistical Software*, *33*(1), 1-22.

Frishkoff, G. A., Frank, R. M., Rong, J., Dou, D., Dien, J., & Halderman, L. K. (2007). A framework to support automated classification and labeling of brain electromagnetic patterns. *Computational Intelligence and Neuroscience*, 2007. doi: 10.1155/2007/14567

Gajewski, P. D., & Falkenstein, M. (2013). Effects of task complexity on ERP components in Go/no-go tasks. *International Journal of Psychophysiology*, *87*(3), 273-278. doi:10.1016/j.ijpsycho.2012.08.007

Gibb, B. E., & Tsypes, A. (2019). Using event-related potentials to improve our prediction of suicide risk. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, *4*(10), 854-855. doi: 10.1016/j.bpsc.2019.08.003

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. 2010. *Journal of Machine Learning Research*. 9:249-256

Goldstein, T. R., Merranko, J., Hafeman, D., Gill, M. K., Liao, F., Sewall, C., ... & Birmaher, B. (2022). A risk calculator to predict suicide attempts among individuals with early-onset bipolar disorder. *Bipolar Disorders*, *24*(7), 749-757. doi: 10.1111/bdi.13250

Görtler, J., Hohman, F., Moritz, D., Wongsuphasawat, K., Ren, D., Nair, R., ... & Patel, K. (2022, April). Neo: Generalizing confusion matrix visualization to hierarchical and multi output labels. *CHI Conference on Human Factors in Computing Systems*, 1-13. doi: 10.1145/3491102.3501823

Gratton, G., Coles, M. G., & Donchin, E. (1983). A new method for off-line removal of ocular artifact. *Electroencephalography and Clinical Neurophysiology*, *55*(4), 468-484. doi: 10.1016/0013-4694(83)90135-9

Gvion, Y., & Levi-Belz, Y. (2018). Serious suicide attempts: systematic review of psychological risk factors. *Frontiers in Psychiatry*, *9*, 56. doi: 10.3389/fpsyt.2018.00056/full

Hack, L., Jovanovic, T., Carter, S., Ressler, K., & Smith, A. (2017). 894. Suicide prediction using machine learning techniques in screening and clinician-derived data. *Biological Psychiatry*, *81*(10), S361. doi: 10.1016/j.biopsych.2017.02.619

Haldane, M., Cunningham, G., Androutsos, C., & Frangou, S. (2008). Structural brain correlates of response inhibition in Bipolar Disorder I. *Journal of Psychopharmacology*, *22*(2), 138-143. doi:10.1177/0269881107082955

Hasey, G., Colic, S., Reilly, J., MacCrimmon, D., Khodayari, A., DeBruin, H., & Mistry, N. (2020). Detection of suicidal ideation in depressed subjects using resting electroencephalography features identified by machine learning algorithms. *Biological Psychiatry*, *87*(9), S380-S381. doi: 10.1016/j.biopsych.2020.02.974

Hastie, T., & Qian, J. (2016). Glmnet vignette. *Retrieved June*, *9*(2016), 1-30.

Hastie, T., Tibshirani, R., & Friedman, J. (2017): The elements of statistical learning. 2nd edition. *Springer*. pp.389-400

Hettige, N. C., Nguyen, T. B., Yuan, C., Rajakulendran, T., Baddour, J., Bhagwat, N., Bani-Fatemi, A., Voineskos, A.N., Charkravarty, M. M., & De Luca, V. (2017). Classification of suicide attempters in schizophrenia using sociocultural and clinical features: A machine learning approach. *General Hospital Psychiatry*, *47*, 20-28. doi: 10.1016/j.genhosppsych.2017.03.001.

Hoffer, E., Hubara, I., & Soudry, D. (2017). Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *Advances in Neural Information Processing Systems*, 30. https://dl.acm.org/doi/10.5555/3294771.3294936

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, *36*(3), 1171-1220. doi: 10.1214/009053607000000677

Høyer, E., Mortensen, P., & Olesen, A. (2000). Mortality and causes of death in a total national sample of patients with affective disorders admitted for the first time between 1973 and 1993. *British Journal of Psychiatry, 176*(1), 76-82. doi:10.1192/bjp.176.1.76

Jacobucci, R., & Li, X. (2022). Does Minority Case Sampling Improve Performance with Imbalanced Outcomes in Psychological Research?. *Journal of Behavioral Data Science*, 2(1), 59-74. doi: 0.35566/jbds/v2n1/p3

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning with applications in R, 2nd edition, *Springer.* pp.15-41, pp. 337-358

Jiang, T., Gradus, J. L., & Rosellini, A. J. (2020). Supervised machine learning: a brief primer. *Behavior Therapy*, 51(5), 675-687. doi: 10.1016/j.beth.2020.05.002

Jijkoun, V., & Hofmann, K. (2009, March). Generating a non-english subjectivity lexicon: Relations that matter. *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 398-405.

Jung, J. S., Park, S. J., Kim, E. Y., Na, K. S., Kim, Y. J., & Kim, K. G. (2019). Prediction models for high risk of suicide in Korean adolescents using machine learning techniques. *PloS One*, *14*(6), e0217639. doi: 10.1371/journal.pone.0217639

Kalinowski, T., Falbel, D., Allaire, J.J., Chollet, F., Rstudio, Google, Tang, Y., Bijl, W.B.D., Studer, M., & Keydana, S. (2021): Package 'keras', https://cran.r-project.org/web/packages/keras/keras.pdf

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90. doi: 10.1145/3065386

Kropotov, J. (2016). Functional neuromarkers for psychiatry: Applications for diagnosis and treatment. *Academic Press*. doi: 10.1016/B978-0-12-410513-3.00006-1

Li, Z., Kamnitsas, K., & Glocker, B. (2020). Analyzing overfitting under class imbalance in neural networks for image segmentation. *IEEE transactions on Medical Imaging*, 40(3), 1065-1077. doi: 10.1109/TMI.2020.3046692

Liotti, M., Pliszka, S. R., Higgins, K., Perez III, R., & Semrud-Clikeman, M. (2010). Evidence for specificity of ERP abnormalities during response inhibition in ADHD children: A comparison with reading disorder children without ADHD. *Brain and Cognition*, 72(2), 228-237. doi: 10.1016/j.bandc.2009.09.007

Lydia, A., & Francis, S. (2019). Adagrad-an optimizer for stochastic gradient descent. *International Journal of Information and Computing Science*, *6*(5), 566-568.

Maalouf, M. (2011). Logistic regression in data analysis: an overview. *International Journal of*

Data Analysis Techniques and Strategies, *3*(3), 281-299. doi: 10.1504/IJDATS.2011.041335

Maher, N. A., Senders, J. T., Hulsbergen, A. F., Lamba, N., Parker, M., Onnela, J. P., ... & Broekman, M. L. (2019). Passive data collection and use in healthcare: A systematic review of ethical issues. *International Journal of Medical Informatics*, *129*, 242-247. doi:10.1016/j.ijmedinf.2019.06.015

Mahesh, B. (2018). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*, *9*, 381-386. Doi: 10.21275/ART20203995

McCormick, U., Murray, B., & McNew, B. (2015). Diagnosis and treatment of patients with bipolar disorder: A review for advanced practice nurses. *Journal of the American Association of Nurse Practitioners*, *27*(9), 530-542. doi: 10.1002/2327-6924.12275

McInnis, M. G., Assari, S., Kamali, M., Ryan, K., Langenecker, S. A., Saunders, E. F., Versha, K., Evans, S., O'Shea, K. S., Mower Provost, E., & Marshall, D. (2018). Cohort profile: the Heinz C. Prechter longitudinal study of bipolar disorder. *International Journal of Epidemiology*, *47*(1), 28-28n. doi: 10.1093/ije/dyx229

Meule, A. (2017). Reporting and interpreting task performance in go/no-go affective shifting tasks. *Frontiers in Psychology*, *8*, 701. doi:10.3389/fpsyg.2017.00701

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C., Lin, C. (2021). Package 'e1071', https://cran.r-project.org/web/packages/e1071/e1071.pdf

Miché, M., Studerus, E., Meyer, A. H., Gloster, A. T., Beesdo-Baum, K., Wittchen, H. U., & Lieb, R. (2020). Prospective prediction of suicide attempts in community adolescents and young adults, using regression methods and machine learning. *Journal of Affective Disorders*, *265*, 570-578. doi: 10.1016/j.jad.2019.11.093

Microsoft, & Ooi, H. (2021): Package 'glmnetUtils'. https://cran.r-project.org/web/packages/glmnetUtils/glmnetUtils.pdf

Miller, J.N., & Black, D.W. Bipolar Disorder and Suicide: a Review. *Curr Psychiatry Rep* 22, 6 (2020). doi: 10.1007/s11920-020-1130-0

Moeller, F. G., Barratt, E. S., Dougherty, D. M., Schmitz, J. M., & Swann, A. C. (2001). Psychiatric aspects of impulsivity. *American Journal of Psychiatry*, *158*(11), 1783-1793. doi: 10.1176/appi.ajp.158.11.1783

Moraes, P. H. P. D., Neves, F. S., Vasconcelos, A. G., Lima, I. M. M., Brancaglion, M., Sedyiama, C. Y., ... & Malloy-Diniz, L. F. (2013). Relationship between neuropsychological and clinical aspects and suicide attempts in euthymic bipolar patients. *Psicologia: Reflexão e Crítica*, *26*, 160-167. doi: 10.1590/S0102-79722013000100017

Muth, C., Bales, K. L., Hinde, K., Maninger, N., Mendoza, S. P., & Ferrer, E. (2016). Alternative models for small samples in psychological research: applying linear mixed effects models and generalized estimating equations to repeated measures data. *Educational and Psychological Measurement, 76*(1), 64-87. doi: 10.1177/0013164415580432

Najt, P., Perez, J., Sanches, M., Peluso, M. A. M., Glahn, D., & Soares, J. C. (2007). Impulsivity and bipolar disorder. *European Neuropsychopharmacology*, *17*(5), 313-320. doi: 10.1016/j.euroneuro.2006.10.002

Nakerst, G., Brennan, J., & Haque, M. (2020). Gradient descent with momentum---to accelerate or to super-accelerate? *arXiv preprint arXiv:2001.06472*. doi:10.48550/arXiv.2001.06472

Navarro, M.C., Quellet-Morin, I., Geoffroy, M.C., Boivin, M., Tremblay, R.E., Côté, S.M., & Orri, M. (2021): Machine learning assessment of early life factors predicting suicide attempt in adolescence or young adulthood, JAMA Network Open, 4(3):2211450, doi:10.1001/jamanetworkopen.2021.1450

Novick, D. M., Swartz, H. A., & Frank, E. (2010). Suicide attempts in bipolar I and bipolar II disorder: a review and meta-analysis of the evidence. *Bipolar Disorders*, *12*(1), 1-9. doi: 10.1111/j.1399-5618.2009.00786.x

Nurnberger, J. I., Blehar, M. C., Kaufmann, C. A., York-Cooler, C., Simpson, S. G., Harkavy-Friedman, J., Severe, J. B., Malaspina, D., & & Reich, T. (1994). Diagnostic interview for genetic studies: rationale, unique features, and training. *Archives of General Psychiatry, 51*(11), 849-859. doi: 10.1001/archpsyc.1994.03950110009002

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*. doi: 10.48550/arXiv.1811.03378

Passos, I. C., Mwangi, B., Cao, B., Hamilton, J. E., Wu, M. J., Zhang, X. Y., ... & Soares, J. C. (2016). Identifying a clinical signature of suicidality among patients with mood disorders: A pilot study using a machine learning approach. *Journal of Affective Disorders*, *193*, 109-116. doi: 10.1016/j.jad.2015.12.066

Patel, S. H., & Azzam, P. N. (2005). Characterization of N200 and P300: selected studies of the event-related potential. *International Journal of Medical Sciences*, *2*(4), 147-154. doi: 0.7150/ijms.2.147

Putnick, D. L., & Bornstein, M. H. (2016). Measurement invariance conventions and reporting: The state of the art and future directions for psychological research. *Developmental Review, 41*, 71-90. doi; 10.1016/j.dr.2016.06.004

Ramraj, S., Uzir, N., Sunil, R., & Banerjee, S. (2016). Experimenting XGBoost algorithm for prediction and classification of different datasets. *International Journal of Control Theory and Applications*, *9*(40).

R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, http://www.R-project.org/

Scharf, F., Widmann, A., Bonmassar, C., & Wetzel, N. (2022). A tutorial on the use of temporal principal component analysis in developmental ERP research–Opportunities and challenges. *Developmental Cognitive Neuroscience*, 101072. doi: 10.1016/j.dcn.2022.101072

Seo, J. H., & Kim, Y. H. (2018). Machine-learning approach to optimize smote ratio in class imbalance dataset for intrusion detection. *Computational Intelligence and Neuroscience*, doi: 10.1155/2018/9704672

Septiadi, J., Warsito, B., & Wibowo, A. (2020). Human Activity Prediction using Long Short Term Memory. *E3S Web of Conferences*, Vol. 202, 15008, *EDP Sciences*. doi: 10.1051/e3sconf/202020215008

Sharma, R., & Markar, H. R. (1994). Mortality in affective disorder. *Journal of Affective Disorders*, *31*(2), 91-96. doi: 10.1016/0165-0327(94)90112-0

Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Science*, *6*(12), 310-316.

Shen, L., Er, M. J., & Yin, Q. (2022). Classification for High-dimension Low-Sample Size Data. *Pattern Recognition*, 108828. doi: 10.1016/j.patcog.2022.108828

Simpson, S. G., & Jamison, K. R. (1999). The risk of suicide in patients with bipolar disorders. *Journal of Clinical Psychiatry*, *60*(2), 53-56.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of Machine Learning Research*, *15*(1), 1929-1958. doi:10.1038/s41398-020-01100-0

Srivastava, R. K., Masci, J., Kazerounian, S., Gomez, F., & Schmidhuber, J. (2013). Compete to compute. *Advances in Neural Information Processing Systems*, *26*.

Su, C., Aseltine, R., Doshi, R., Chen, K., Rogers, S. C., & Wang, F. (2020). Machine learning for suicide risk prediction in children and adolescents with electronic health records. *Translational Psychiatry*, *10*(1), 1-10. doi:10.1038/s41398-020-01100-0

Swann, A. C., Dougherty, D. M., Pazzaglia, P. J., Pham, M., Steinberg, J. L., & Moeller, F. G. (2005). Increased impulsivity associated with severity of suicide attempt history in patients with bipolar disorder. *American Journal of Psychiatry*, *162*(9), 1680-1687. doi: 10.1176/appi.ajp.162.9.1680

Swann, A. C., Lijffijt, M., Lane, S. D., Steinberg, J. L., & Moeller, F. G. (2009). Severity of bipolar disorder is associated with impairment of response inhibition. *Journal of Affective Disorders*, *116*(1-2), 30-36. doi: 10.1016/j.jad.2008.10.022

Tavakoli, P., Boafo, A., Jerome, E., & Campbell, K. (2021). Active and passive attentional processing in adolescent suicide attempters: An event-related potential study. *Clinical EEG and Neuroscience*, *52*(1), 29-37. doi:10.1177/155005942093308

Torgo, L. (2013). Package 'DMwR', https://www2.uaem.mx/r-mirror/web/packages/DMwR/DMwR.pdf

Vabalas, A., Gowen, E., Poliakoff, E., & Casson, A. J. (2019). Machine learning algorithm validation with a limited sample size. *PloS One*, *14*(11), e0224365. doi: 10.1371/journal.pone.0224365

Verbruggen, F., & Logan, G. D. (2008). Automatic and controlled response inhibition: associative learning in the go/no-go and stop-signal paradigms. *Journal of Experimental Psychology: General*, *137*(4), 649. doi: 10.1037/a0013170

Weisbrod, M., Kiefer, M., Marzinzik, F., & Spitzer, M. (2000). Executive control is disturbed in schizophrenia: evidence from event-related potentials in a Go/no-go task. *Biological Psychiatry*, *47*(1), 51-60. doi: 10.1016/S0006-3223(99)00218-8

Yeom, S., Giacomelli, I., Fredrikson, M., & Jha, S. (2018, July). Privacy risk in machine learning: Analyzing the connection to overfitting. *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 268-282. IEEE. doi:10.1109/CSF.2018.00027

Ying, X. (2019, February). An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, Vol. 1168, No. 2, 022022. IOP Publishing. doi:10.1088/1742-6596/1168/2/022022

Zeiler, M. D., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q. V., ... & Hinton, G. E. (2013, May). On rectified linear units for speech processing. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 3517-3521. IEEE. doi:10.1109/ICASSP.2013.6638312

Zhang, G., Wang, C., Xu, B., & Grosse, R. (2018). Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*.

Zhang, Y., & Yang, Y. (2015). Cross-validation for selecting a model selection procedure. *Journal of Econometrics*, *187*(1), 95-112. doi:10.1016/j.jeconom.2015.02.006

Zhang, Z. (2018). Improved adam optimizer for deep neural networks. *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 1-2. IEEE. doi: 10.1109/IWQoS.2018.8624183

Zhang, Z., & Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems*, *31*. doi: 10.5555/3327546.3327555

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (statistical methodology)*, 67(2), 301-320. doi: 10.1111/j.1467-9868.2005.00503.x