Noname manuscript No.
(will be inserted by the editor)

Imbalanced Learning with Parametric Linear Programming Support Vector Machine for Weather Data Application

Elaheh Jafarigol* · Theodore B. Trafalis

Received: date / Accepted: date

Abstract Imbalanced learning is an aspect of predictive modeling and machine learning that has taken a lot of attention in the last decade. Due to the nature of rare events, finding a reliable and efficient classification method for imbalanced data set has been challenging, and multiple research projects have been carried out to improve the existing algorithms for more accurate predictions of such data sets. To this end, we propose a Linear Programming Support Vector Machine (LP-SVM) applicable to imbalanced data. Apart from model selection and modifications, we have also implemented a parameter selection method based on the parametric simplex approach for parameter tuning of LP-SVM. For numerical tests, we have used a real data set consisting of weather observations made by Bureau of Meteorology's (BM) system in Australia, and the results show that the proposed method works pretty well on the tested examples.

Keywords Imbalanced learning · classification · support vector machine · parameter tuning · machine learning

1 Introduction

In recent years, the use of machine learning, and data mining techniques have gained a lot of attention from the data analytics society. The generality and

E. Jafarigol
School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK 73019, USA
Tel.: +1405-445-9829
E-mail: elaheh.jafarigol@ou.edu

T.B. Trafalis
School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK 73019, USA
Tel.: +1405-325-3721
E-mail: ttrafalis@ou.edu

functionality of these techniques have made them applicable in various disciplines, from industry to academia. Imbalanced learning is the art of retaining knowledge from a data set where the number of instances in one class called the majority class is significantly larger than the number of instances in the other class known as the minority class. The minority class consists of rare cases that are more important from the learning perspective. The main concern in imbalanced learning is that most classifiers are biased towards the majority class, and despite having high accuracy score, they fail to classify the instances in the minority class correctly. Therefore, the misclassification error of the minority class is often ignored [**?**]. Throughout the years, multiple research projects with a focus on imbalanced learning have been carried out. However, none of these approaches has reached optimization yet, and the need for more powerful algorithms still exists [**?**] [**?**] [**?**]. The use of machine learning algorithms in weather applications was initiated to improve the warnings' lead time to critical weather phenomena such as thunderstorms and tornadoes. In the research carried out by Marzban and Stumpf [**?**] the Mesocyclone Detection Algorithm (MDA) attributes based on Doppler radar observations were used to train artificial neural network algorithms and predict tornadic events. Later, Lakshmanan combined MDA data with near-storm environment (NSE) data to improve the prediction results [**?**].

Many weather events such as rain are considered as rare events in some locations, and various algorithms such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM) were used to improve the accuracy of predictions in the minority class [**?**]. Combination of machine learning algorithms with sampling or cost-sensitive methods has led to an improvement in classification results [**?**]. An important issue with imbalanced learning is that the standard evaluation metrics cannot provide a concise assessment of both classes. Shaza et al. [**?**] have provided an extensive review of different evaluation metrics developed for imbalanced learning. In this paper, multiple visualization tools such as Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) as a means of performance assessment are also discussed. To explore the distinctive features of imbalanced data sets, Lopez et al. [**?**] did extensive work on using the intrinsic characteristics and their related issues with classification. A popular approach to imbalanced learning is SVM, as discussed by Trafalis et al. [**?**]. Comparison of SVM and ANNs proved that SVM and its modified versions such as Bayesian SVM and SVM with Recursive Feature Elimination (RFE) with threshold adjustment on SVM could outperform ANN for imbalanced learning. Other variations of SVM, as discussed in [**?**, **?**] is introduced as a replacement for quadratic SVM in case of redundant and noisy data. In this regard, linear and kernel SVM with $L_1 - norm$ objective function is also introduced to be compared with quadratic SVM [**?**]. In [**?**], a multi-objective approach that incorporates an individual objective function for the positive and negative error sums of minority and majority class is introduced. The three-objective optimization model coupled with $L_1 - norm$ SVM presents a significant improvement in the performance of the classifier. Principles of multi-objective optimization have been widely implemented in

several applications [**?**, **?**, **?**]. The primary objective of this paper is to provide a reliable approach that can efficiently and accurately classify imbalanced data sets encountered in real-life and large-scale problems. To this end, a two objective linear programming SVM is defined to enrich the classification algorithm using a parametric simplex method to tune the parameters based on an algorithmic approach. The mathematics behind the parametric simplex algorithm for multi-objective optimization in this work is initially discussed by Ehrgott et al. [**?**]. In the next section, an overview of the dataset and relevant preprocessing methods in addition to the mathematical setup of the methodology is provided. In Section 3, the numerical results are discussed followed by the limitations and suggested future work. The paper is concluded in Section 4.

2 Material and methods

2.1 Dataset and Preprocessing the Data

The data used in this work is collected by the BM in Australia, which is responsible for collecting the data regarding weather phenomena in Australia. BM weather stations collect data from real-time observations, including temperature, rainfall, wind speed and direction, sunshine, etc. The observations are recorded in different time intervals such as daily, or hourly, and the observation procedure varies based on the location of the station and the weather phenomena observed. Due to limitations of CPU and Memory, the data derived from 15 stations with 19 attributes is divided into 3 sample groups. Each group is treated as one data set and is used for analysis. Fig. 1 presents an overview of the percentage of observations in each class.

Before proceeding with the analysis, it is essential to check the data for missing values and outliers. The attributes with more than 30 missing values in each sample are dropped, and the remaining missing values are substituted with mean value imputation. As it is shown in Fig. 2, the number of features in the data set is reduced to 14, which includes redundant and noisy data, so feature selection methods are used to reduce the sample size and find the optimal number of features.

RFE with 5-fold Cross Validation (CV) is a standard feature selection approach. We have used RFE with Random Forest to find the optimal number of features for classification, and the results are presented in Fig. 3.

In addition to the optimal number of features, it is essential to consider how the features contribute to a more precise classification result. Random forest feature importance plot presented in Fig. 4 provides an insight into the importance of features in the data set.

Based on the results presented in Fig. 3 and 4, respectively, the four least important features were dropped. The list of most important features for classification is presented in table 1.

After selecting the optimal feature set, Principal Component Analysis (PCA) is implemented to construct the feature set that captures 99% of the

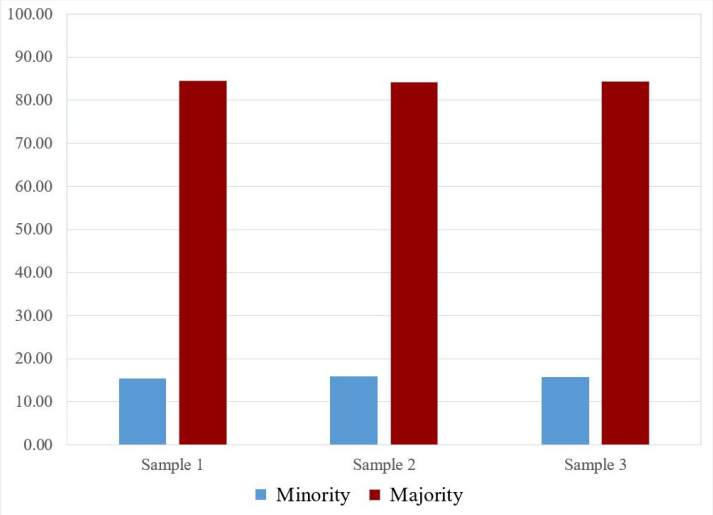


Fig. 1: Data Distribution

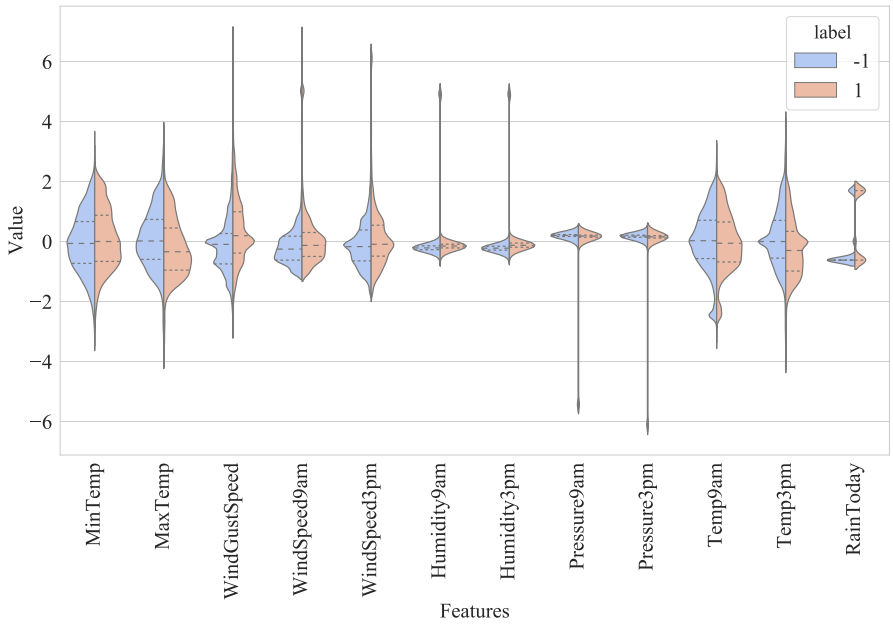


Fig. 2: Visual presentation of data based on features, in each class

variance. Based on the results presented in Fig. 5, PCA has reduced the number of features to 6 principal components, while 99% of the variance is captured.

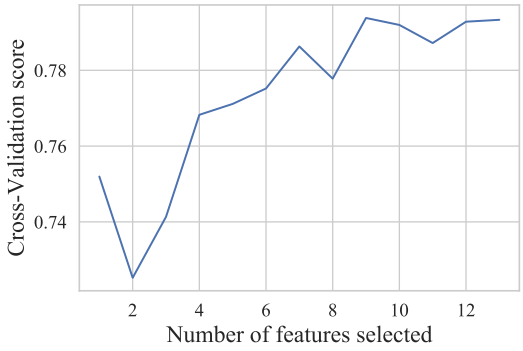


Fig. 3: RFE-RF with 5-fold cross validation

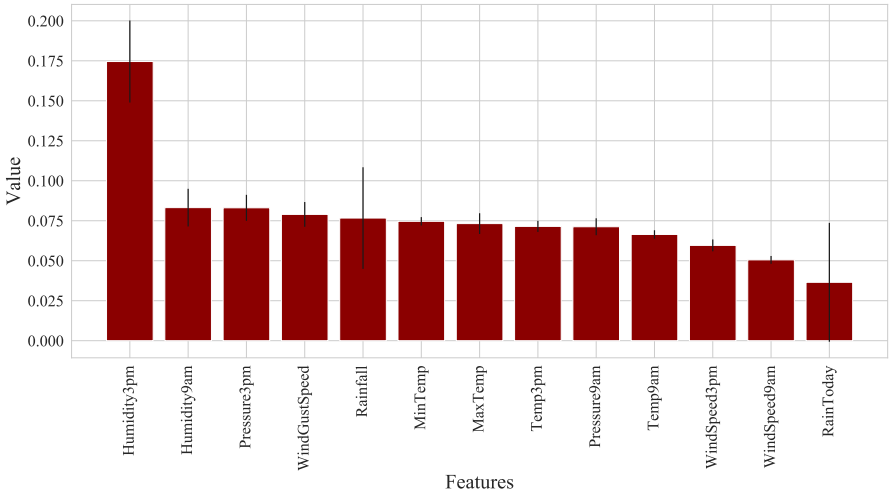


Fig. 4: Feature Importance plot

When dealing with imbalanced data, resampling methods can enhance the performance of the classifier [**?**]. Generally, resampling methods follow two strategies; one is removing instances from the majority class known as under-sampling and second is adding new instances to the minority class, which is known as over-sampling. In the final preprocessing step of this work, we have employed an over-sampling method known as Synthetic Minority Over-sampling Technique (SMOTE), which is implemented with Imbalanced-learn API package in Python, that provides fast and accurate sampling strategies for imbalanced data.

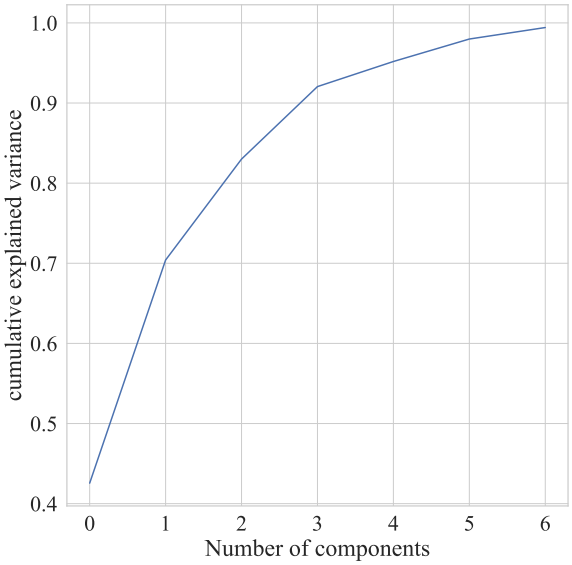


Fig. 5: Optimal Number of Features

2.2 SVM, Concepts and Classical Methods

SVM is a popular classification algorithm with applications in fraud detection, identifying cancer cells, face recognition, weather predictions, etc. SVM is a

Table 1: List of Features with their Ranks.

Rank	Feature Name	Meaning	Unit
1	Humidity3pm	Relative humidity at 3 PM	Percent
2	Humidity9pm	Relative humidity at 9 AM	Percent
3	WindGustSpeed	Speed of the strongest wind gust in the 24 hrs	Kilometers
4	Pressure3pm	Atmospheric pressure reduced to mean sea level (3 PM)	Hectopascals
5	Rainfall	Precipitation in the 24 hrs to 9 AM	Millimeters
6	MinTemp	Minimum temperature in the 24 hrs to 9 AM	Degrees Celsius
7	Pressure9am	Atmospheric pressure reduced to mean sea level (9 AM)	Hectopascals
8	Temp3pm	Temperature at 3 PM	Degrees Celsius
9	MaxTemp	Maximum Temperature in the 24 hrs to 9 AM	Degrees Celsius

supervised learning method, developed to find a binary classifier based on training data. There are multiple variations of SVM in the literature, but binary SVM is the most popular one [**?**, **?**]. SVM classifies the data by finding the separating hyperplane denoted as H(x). H(x), also known as the decision function, is the set of all points that satisfy condition (1):

$$\begin{aligned} &H(x) = w^T x + b = 0 \\ &x, w \in \mathbb{R}^n \; and \; b \in \mathbb{R} \end{aligned} \tag{1}$$

where **x** is the vector of features, **w** is the weight vector, and b is the offset. In this formulation $\hat{y} = sign(H(x))$ determines how the test data is classified into two classes using condition (2):

$$\hat{y} = \begin{cases} 1 & w^T x + b > 0 \\ -1 & w^T x + b < 0 \end{cases} \tag{2}$$

The distance from the nearest data point in the training set to the separating hyperplane determines the margin of separation. The hyperplane with the maximum margin of separation is unique and is identified through optimization. SVM optimization problem as introduced in [**?**] is a convex quadratic minimization problem which means local minimum do not exist, and the optimization problem is feasible. A drawback of this approach is that most real-life, large-scale data sets are non-linearly separable, and perfect separation of the two classes is not possible; therefore, soft margin SVM is introduced, and the slack variables denoted as ζ_i are added to the formulation to measure the violation from the separating hyperplane. In the formulation corresponding to soft margin SVM, the goal is to maximize the margin of separation, while minimizing the slack variables with the correct regularization parameter. The regularization parameter denoted as C is defined in the objective function to control the trade-off between the two terms, maximizing the margin of separation and minimizing the slack variables that represent the cost of misclassification. The mathematical formulation of soft margin SVM is:

$$\begin{aligned} &\text{Min}_{w,b,\zeta_i} \left\{ \frac{\|w\|^2}{2} + C \sum_{i=1}^{n} (\zeta_i)^k \right\} \\ &\text{Subject to: } y_i \left(w^T x_i + b \right) \geq 1 - \zeta_i \\ &\zeta_i \geq 0, \forall x_i \in D \end{aligned} \tag{3}$$

Where D is the set of training input vectors. Unfortunately, SVM, as discussed, does not yield the most accurate results, so kernel SVM have been developed to improve the performance of the classifier. Kernelization has been employed to enhance the performance of various machine learning (ML) algorithms. Comparing to non-kernelized SVM, kernel-SVM can provide better

generalization, and it can classify non-linearly separable data more accurately. Different kernel functions have been developed and utilized over the years. Some of the most popular ones are presented in Table 2.

To incorporate the kernel matrix into SVM formulation, the decision function and the constraints must be reformulated as it is shown in (4).

$$\begin{aligned} &\text{Min}_{w,b,\zeta_i}\left\{\frac{\|w\|^2}{2}+C\sum_{i=1}^{n}\left(\zeta_i\right)^k\right\} \\ &\text{Subject to: } y_i\left[\sum_{i=1}^{l}\sum_{j=1}^{l}w_i,k\left(x_i,x_j\right)+b\right]\geq 1-\zeta_i \\ &\zeta_i\geq 0,\forall x_i,x_j\in D \end{aligned} \tag{4}$$

where l is the number of training data points. The updated formulation reflects the use of kernel and transformation of the data from input into the feature space.

2.3 Parametric Linear Programming SVM as a solution algorithm

In this section, we have formulated a modified SVM where the quadratic objective function is replaced with a linear function. To this end, we have utilized a famous metric known as -L_1-norm or mean norm- defined as the sum of the absolute values of the vector components: $\|\overrightarrow{v}\|_1=\sum_{i=1}^{m}|v_i|$. For the weight vector $\mathbf{w}=(w_1,w_2,...,w_n)$ where $n\in\mathbb{N}$ is the size of the feature set and $l\in\mathbb{N}$ is the number of observations, the mathematical formulation of SVM with an L_1-norm objective function is presented below:

Table 2: Kernel Functions

Kernel	Formulation
Linear	$\mathbf{x}^{\mathrm{T}}\mathbf{y}$
Gaussian	$\mathrm{K(x,y)}=\exp\left\{\frac{-\|\mathrm{x}-\mathrm{y}\|^2}{2\delta^2}\right\}$
Polynomial	$\mathrm{K(x,y)}=\left(\mathrm{c}+\mathrm{x}^{\mathrm{T}}\mathrm{y}\right)^{\mathrm{q}}$

$$\begin{aligned}
&\text{Min}_{w,b,\zeta}\left\{\lambda_1 \sum_{i=1}^{l} |w_i| + \lambda_2 \sum_{i=1}^{n} \zeta_i\right\} \\
&\text{Subject to:} \quad y_i\left(w^T x_i + b\right) \geq 1 - \zeta_i \\
&\zeta_i \geq 0, \forall x_i \in D \\
&\lambda_1 + \lambda_2 = 1 \\
&\lambda_1, \lambda_2 \geq 0
\end{aligned} \tag{5}$$

Based on this formulation, that we will refer to as Parametric Linear Programming Support Vector Machine (LP-SVM), the problem can be interpreted as a multi-objective parametric linear programming problem. Therefore, we can employ the widely-used concepts of multi-objective optimization to solve this problem.

In order to obtain the classifier, we have updated the formulation to adopt kernelization. The mathematical formulation of parametric LP-SVM where w_i is the i^{th} component of the weight vector, and $l \in \mathbb{N}$ is the number of observations in the data set, with a given kernel $k(x_i, x_j)$ is presented as follows:

$$\begin{aligned}
&\text{Min}_{w,b,\zeta}\left\{\lambda_1 \sum_{i=1}^{l} |w_i| + \lambda_2 \sum_{i=1}^{n} \zeta_i\right\} \\
&\text{Subject to: } y_i\left[\sum_{i=1}^{l}\sum_{j=1}^{l} w_i, k(x_i, x_j) + b\right] \geq 1 - \zeta_i \\
&\zeta_i \geq 0, \forall x_i, x_j \in D \\
&\lambda_1 + \lambda_2 = 1 \\
&\lambda_1, \lambda_2 \geq 0
\end{aligned} \tag{6}$$

Despite being a powerful algorithm for predictive analysis, implementing SVM can be challenging as the model parameters must be tuned in order to have accurate results, and among various parameter tuning approaches a grid search is commonly used for this purpose [**?**]. To employ a grid search, the model is trained over a set of given values. The optimal parameter value is the one that produces the highest accuracy score. Hence, it is computationally expensive, and it requires powerful computing systems when the data set is large. In this paper, we propose an algorithmic approach based on the parametric simplex algorithm to exhaustively search the solution path and find the optimal value of the regularization parameter.

Note that since the modified SVM proposed in this work is a parametric linear programming problem, we can use the parametric simplex method to attain the potential parameter values and choose the optimal value from a finite set. Various versions of the parametric simplex method exist. These

methods mainly differ in the variable that determines the next iteration [?,?,?]. In this paper, we employ the parametric simplex algorithm for multi-objective optimization problems to find the optimal solution to the two-objective linear programming problem.

The objective function in this problem is a weighted sum of two separate objective functions that we are trying to minimize simultaneously. The goal is to find the optimal values of λ_1 and λ_2 . For simplification, We set $\lambda_1 = \lambda$ and $\lambda_2 = l - \lambda$. The algorithm starts with the feasibility evaluation of the LP problem. To this end, the augmented form of the problem is solved to obtain the optimal basis and optimal basic feasible solution. The augmented form of the problem is given as:

$$\begin{aligned} &\text{Min}_{w,b,\zeta} \left\{ \lambda \sum_{i=1}^{l} |w_i| + (1-\lambda) \sum_{i=1}^{n} \zeta_i \right\} \\ &\text{Subject to: } y_i \left[\sum_{i=1}^{l} \sum_{j=1}^{l} w_i.k\left(x_i, x_j\right) + b \right] - z_i = 1 - \zeta_i \\ &\zeta_i \geq 0, \forall x_i \in D \\ &\lambda \geq 0 \end{aligned} \tag{7}$$

where z_i is the added i_{th} slack variable. The problem is implemented and solved with Gurobi optimization tool in Python. For the cost vector $c \in \mathbb{R}^n$, the two-objective cost function can be written as:

$$\text{Min} \left\{ \left(c^1\right)^T \sum_{i=1}^{l} w_i, \left(c^2\right)^T \sum_{i=1}^{l} \zeta_i \right\} \tag{8}$$

The optimal solution to this problem subject to the constraints would yield the solution to the two-objective LP. The parametric simplex algorithm is implemented in 3 phases.

Phase 1: The augmented LP problem is solved, and the optimal basis and optimal basic feasible solution is obtained. The problem is infeasible if the set of optimal basic feasible solution is empty. Therefore, the algorithm stops.

Phase 2: Starting from the basis obtained in phase one, and for $\lambda = l$, the problem is solved. In this phase, the optimal basis is attained, and the optimal values of the decision variables are updated accordingly.

Phase 3: The set of non-basic variables, denoted by N, and the reduced cost vector for each objective is calculated. The new λ is calculated based on the following equation:

$$\begin{aligned} &\lambda = \max \frac{-\bar{c}_i^2}{\bar{c}_i^1 - \bar{c}_i^2} \\ &\text{for } i \in I, \text{ where } I = \left\{ i \in N, \bar{c}_i^2 < 0, \bar{c}_i^1 > 0 \right\} \end{aligned} \tag{9}$$

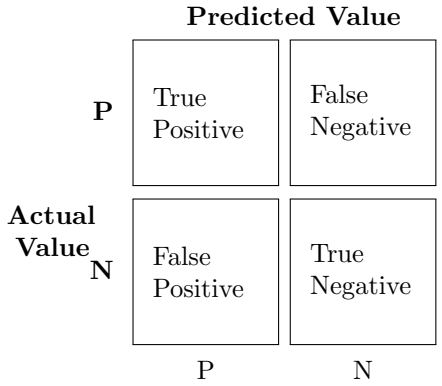
Based on (9) the value of λ and the decision variables is updated after each iteration until $I = \varnothing$ and the algorithm ends while providing a finite number of λs to be tested as an optimal parameter for the LP. This method can significantly reduce the computational complexity and provide more accurate results for parameter tuning in comparison with conventional methods such as a grid search. To recap, the algorithm is implemented using Gurobi in Python, and it is provided as follows:

Algorithm 1 Parameter Selection based on Parametric Simplex Algorithm

Require: model.params.method = 1 {Solver is set to Simplex.}
  model.setObjective (0.0)
  Solve the auxiliary LP
  return Basis
  return Basic feasible solution
  Basis = model.VBasis
  if $Z_i = 0$ **then**
    The LP problem is feasible.
  else
    The LP problem is unbounded.
    Stop the algorithm.
  end if
  Set $\lambda = 1$
  for j = Number of components of the objective function **do**
    Reduced cost of **w**: $\bar{c}_i^2 = w[j].RC$
    Reduced cost of ζ: $\bar{c}_i^2 = \zeta[j].RC$
    N=The set of non-basic variables
    $I = \{i \in N, \bar{c}_i^2 < 0, \bar{c}_i^1 > 0\}$
    while $i \in I, I \neq \varnothing$ **do**
      PStart = Basis {The basis is updated.}
      Solve the auxiliary LP
      return Basis
      return $\lambda = \max \frac{-\bar{c}_i^2}{\bar{c}_i^1 - \bar{c}_i^2}$
    end while
  end for

2.4 Evaluation Metrics

To properly assess the classification results, the metrics specifically defined for imbalanced learning are used. Before further discussing the evaluation metrics, four key terms need to be defined. For a binary classification problem, if $\{P, N\}$ is the predicted labels for the points in the test set, the majority class is denoted by N, and the minority class is denoted by P. A Confusion Matrix as presented is a visual presentation of classifier's performance:



Based on this notation, accuracy and error rate are defined as:

$$\text{Accuracy } = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{10}$$

$$\text{Error rate } = 1 - \text{ accuracy} \tag{11}$$

Although these metrics are commonly used in the literature, it is intuitively perceivable that they are not appropriate for imbalanced data because they are biased towards the majority class and often fail to present the inadequacy of the classifier for imbalanced data. Thus, for binary classification problems of imbalanced data, other metrics are defined and used.

$$\text{Precision } = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

$$\text{Recall } = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

$$\text{F-measure } = \frac{(1+\beta)^2.\ \text{Recall.\ Precision}}{\beta^2.\ \text{Recall } +\ \text{Precision}} \tag{14}$$

where β is the relative importance of precision versus recall, and it is usually set equal to one. Another metric used is the G-mean that is defined as follows:

$$\text{G-mean } = \sqrt{\frac{\text{TP}}{\text{TP} + \text{FN}} \cdot \frac{\text{TN}}{\text{TN} + \text{FP}}} \tag{15}$$

Precision measures the proportion of instances that were labeled correctly among those with the positive label in the test data. In other words, it measures how exact the model is, concerning the minority class. While, recall measures the portion of positive instances in the test data that were labeled correctly, and it measures the completeness of the model, precision and recall have an inverse relationship. Precision is dependent on the distribution of the data, but the recall is more robust. Precision and recall, when used together, can provide a valid insight into the performance of the classifier with respect to

the minority class. Therefore, F-measure is a valuable evaluation metric in imbalanced learning, which can assess the trade-off between precision and recall. Geometric-mean (G-mean) is the metric that is explicitly used for imbalanced learning. By improving G-mean, we try to increase the model's accuracy over each class by considering both classes for evaluation. In this paper, we specifically focus on F-measure and G-mean to compare the models and select the best one.

To visualize and summarize the classifier's performance, we have used Receiver Operating Characteristic curve, also known as ROC curve [**?**]. ROC curve is a popular tool for presenting the trade-off between true positive rate (TP˙Rate) and false positive rate (FP˙Rate) defined as:

$$\begin{aligned} \mathrm{TP}_{\mathrm{Rate}} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\ \mathrm{FP}_{\mathrm{Rate}} &= \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \end{aligned} \tag{16}$$

The area under ROC, denoted as AUC is defined as:

$$\mathrm{AUC} = \frac{\mathrm{TP}_{\mathrm{Rate}} + \mathrm{TN}_{\mathrm{Rate}}}{2} \tag{17}$$

AUC measures the probability of correctly classifying positive instances while minimizing the number of false positives, and it is a great method of comparison between different models as it is independent of the classification model. Fig. 6 shows all the key points in the ROC curve and AUC.

To evaluate the efficiency of classification methods, 10-fold cross validation is integrated within the model as a standard evaluation technique.

In addition to the traditional imbalanced learning evaluation metrics, we have computed the scalar forecast evaluation scores commonly used in meteorology, such as Critical Success Index (CSI) $= \frac{\mathrm{TP}}{\mathrm{TP+FP+FN}}$, Probability of Detention (POD) $= \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, False Alarm Ratio (FAR) $= \frac{\mathrm{FP}}{\mathrm{TP+FP}}$, Bias $= \frac{\mathrm{TP+FP}}{\mathrm{TP+FN}}$ and and Heidke Skill Score (HSS) $= \frac{2*((\mathrm{TP}*\mathrm{TN})-(\mathrm{FN}*\mathrm{FP}))}{(\mathrm{TP+FN})(\mathrm{FN+TN})+(\mathrm{TP+FP})(\mathrm{FP+TN}))}$.

The POD and FAR range between 0 and 1. The POD computes the ratio of forecasted weather events that were predicted correctly while FAR measures the ratio of falsely predicted ones. The optimal value of Bias is one, which shows that the system does not over-forecast or under-forecast the target. HSS which ranges between -1 and 1, with 1 being the best value, evaluates the overall quality of predictions [**?**, **?**]. Analysis of the forecast evaluation scores allows us to make a meaningful interpretation of the experimental results considering the model's application in the prediction of weather phenomena.

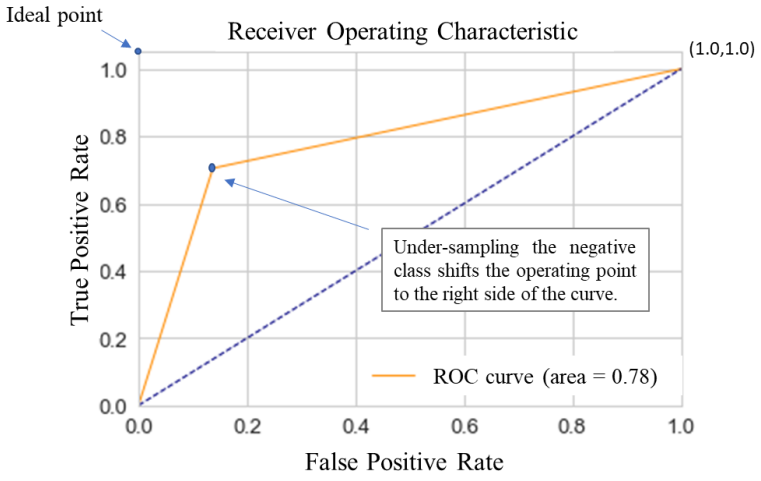


Fig. 6: Receiver Operating Characteristic and Area Under the Curve

3 Results and Discussion

3.1 Parameter Selection Results

In this section, a comparison of standard SVM and parametric LP-SVM is presented. The proposed parametric LP-SVM is implemented in Python using Gurobi Optimization tool. The model is tuned based on the parameter obtained from the state-of-the-art parametric modeling algorithm discussed in section 2.3, and the optimal value of $\lambda = 0.9999$ is attained. The objective function of the optimization problem is updated to:

$$\text{Min}_{w,b,\zeta} \left\{ 0.9999 \sum_{i=1}^{l} |w_i| + 0.0001 \sum_{i=1}^{n} \zeta_i \right\} \tag{18}$$

To implement standard SVM, we have used available classification packages in Python. Incorporating $C = \frac{\lambda_1}{\lambda_2}$ in formulation (4) the value of $C = 0.0001$ is set as the regularization parameter based on the value obtained from the parametric simplex algorithm.

3.2 Numerical Results

To attain numerical results, we run both classifiers on three samples with 10-fold CV. For kernel SVM using RBF, the parameter $\gamma = 0.1$ is found based on a grid search over a set of $\gamma = \{10, 1, 0.1, 0.01, 0.001, 0.00001\}$. The scores obtained from both algorithms are presented in Table 3,4,5.

With G-mean = 0, it is plain to perceive that Gaussian kernel does not provide our desired generalization to be able to classify both classes, as well as polynomial kernel which fails to provide acceptable prediction results. An

Table 3: Numerical results, Group 1.

Model Type	Kernel	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Standard SVM	Linear	0.67	0.79	0.47	0.59	0.64	0.71
	Gaussian	0.49	0.25	0.50	0.65	0.00	NAN
	Polynomial	0.51	0.89	0.01	0.08	0.12	0.70
LP-SVM	Linear	0.67	0.71	0.57	0.63	0.66	0.67
	Gaussian	0.50	0.50	1.00	0.67	0.00	NAN

Table 4: Numerical results, Group 2.

Model Type	Kernel	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Standard SVM	Linear	0.64	0.78	0.40	0.53	0.60	0.69
	Gaussian	0.49	0.24	0.50	0.66	0.00	NAN
	Polynomial	0.51	0.90	0.07	0.08	0.16	0.70
LP-SVM	Linear	0.66	0.72	0.51	0.60	0.64	0.67
	Gaussian	0.50	0.50	1.00	0.67	0.00	NAN

Table 5: Numerical results, Group 3.

Model Type	Kernel	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Standard SVM	Linear	0.61	0.72	0.44	0.53	0.60	0.69
	Gaussian	0.49	0.24	0.50	0.66	0.00	NAN
	Polynomial	0.51	0.90	0.07	0.08	0.16	0.70
LP-SVM	Linear	0.66	0.72	0.51	0.60	0.64	0.67
	Gaussian	0.50	0.50	1.00	0.67	0.00	NAN

interesting observation can be made that in the comparison between three commonly used kernels, a linear kernel which is the least computationally challenging kernel function, provides enough generalization to perform equally well on both classes and it generates the highest scores.

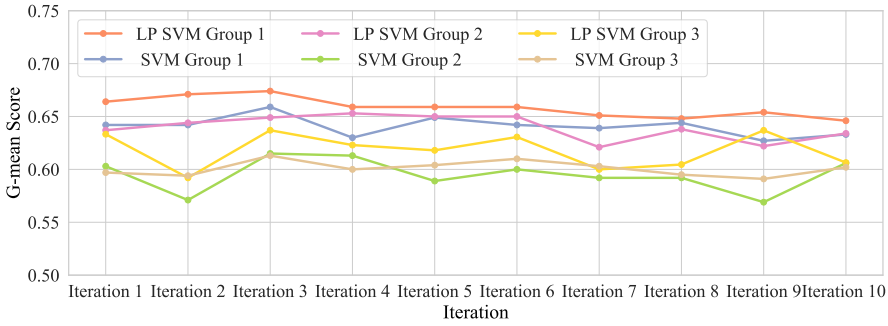


Fig. 7: G-mean Score with 10-fold CV

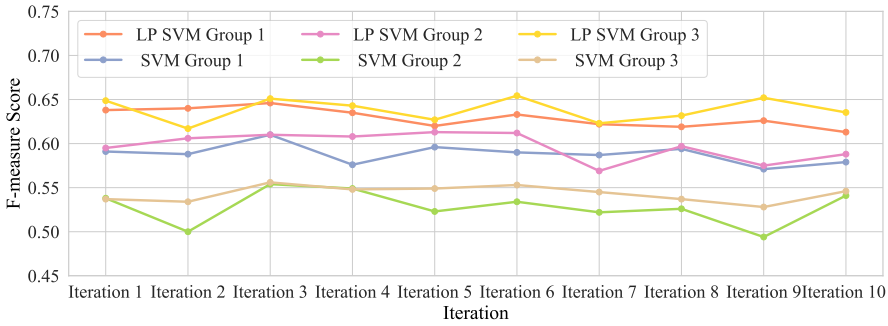


Fig. 8: F-measure Score with 10-fold CV

By looking at F-measure and G-mean from figures 7 and 8 at each iteration, it can be inferred that the proposed LP-SVM performs better in comparison with SVM. These two metrics are especially important because they consider both classes in performance evaluation. As it is mentioned before, the meaningful range of AUC score is between 0.5 and 1, and the closer we get to 1, the better the classifier is. This score is important because it also determines the probability of ranking an unknown instance positive, which in the case of our work, is the probability of having rain. So, the probability of predicting a randomly selected instance as positive is approximately 0.7 in different samples, which is a reasonable result for an imbalanced learning problem.

It is worth mentioning that, to improve the results of cross-validation, we have randomized the data before using 10-fold CV because randomization before classification ensures that observations from both classes exist in all iterations. As for other metrics, having the relatively close performance of the classifier for both classes is an indication that the minority class is not ignored, and without loss of generalization, this approach is a suitable method for imbalanced data sets.

For further evaluation of the proposed model, we have calculated the scalar forecast evaluation scores, and we specifically focus on POD and FAR, which evaluate the quality of predictions [**?**, **?**]. By comparing the results in Tables 6, 7 and 8 it can be observed that the highest POD and FAR scores are attained from classifying the data with LP-SVM.

3.3 Limitations and Future work

Despite the advances made in data analytics, there are still various opportunities for research projects to be carried out to provide insight into ML algorithms. Large scale problems are becoming more popular, and we will con-

Table 6: Scalar Forecast Scores, Group 1.

Model Type	Kernel	CSI	POD	FAR	Bias	HSS
Standard SVM	Linear	0.57	0.62	0.12	0.71	0.35
	Gaussian	0.25	0.49	0.50	0.49	0.00
	Polynomial	0.50	0.50	0.00	0.50	0.01
LP-SVM	Linear	0.53	0.64	0.23	0.83	0.33
	Gaussian	0.00	NAN	1.00	NAN	0.00

Table 7: Scalar Forecast Scores, Group 2.

Model Type	Kernel	CSI	POD	FAR	Bias	HSS
Standard SVM	Linear	0.54	0.60	0.11	0.67	0.29
	Gaussian	0.24	0.49	0.50	0.49	0.00
	Polynomial	0.50	0.51	0.00	0.51	0.02
LP-SVM	Linear	0.54	0.62	0.19	0.77	0.31
	Gaussian	0.00	NAN	1.00	NAN	0.00

tinue to face the challenges associated with them. For the parametric simplex method introduced for parametric modeling, we propose implementing this method for identifying the hyperparameters in different algorithms other than SVM, as it can reduce the computational complexity and effectively improve the parameter tuning in statistical learning.

As for imbalanced learning, more accurate methods regarding the minority class are still required. To this end, we tend to explore how the proposed algorithmic approach can work in combination with data-related methods to improve classification effectiveness and efficiency. we also suggest adding separate misclassification costs for minority and the majority class as discussed in [**?**], which can be expected to further improve the results for both classes.

4 Conclusion

Based on the discussion of the model and review of the relevant literature, we can claim that parametric LP-SVM is a promising method for classification of imbalanced data. As it is previously discussed, modification of SVM to a two-objective parametric LP problem, allows us to benefit from multi-criteria optimization and employ a parametric simplex algorithm to attain the optimal tuning parameter. In this article, we tested the proposed method on real-life meteorological observations and the results obtained from training and testing the data prove that the proposed method applies to imbalance datasets. To summarize, the main contributions of this article towards improving the classification of imbalanced data are: 1. Implementing an algorithmic approach for finding the optimal regularization parameter in SVM. This approach is based on parametric simplex optimization that searches the solution path and provides a finite number of values that could be used as a possible regularization parameter. The most important advantage of this method is that it is built upon a solid mathematical background and is considerably faster than grid search. 2. Implementing and Evaluation of an LP-SVM constructed based on the definition of $L_1 - norm$ to effectively predict the minority class as well as

Table 8: Scalar Forecast Scores, Group 3.

Model Type	Kernel	CSI	POD	FAR	Bias	HSS
Standard SVM	Linear	0.53	0.60	0.17	0.72	0.26
	Gaussian	0.20	0.49	0.60	0.49	0.00
	Polynomial	0.49	0.53	0.13	0.57	0.02
LP-SVM	Linear	0.43	0.63	0.43	1.11	0.24
	Gaussian	0.00	NAN	1.00	NAN	0.00

the majority class. Modified SVM can decrease the computational complexity associated with solving a quadratic optimization problem, and it can be solved using the existing software suitable for LP problems.

Compliance with Ethical Standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.