Classification of Drug Effectiveness from Online Customer Reviews

A Comparative Study on Multilayer Perceptron and Support Vector Machine

1 INTRODUCTION

1.1 Description and Problem Motivation

Safety and effectiveness of pharmaceutical product have long been a concerned healthcare issue. The difference in drug reactions is highly subjected to patients' body and treatment conditions. Close monitoring is necessary to support drug products' research and development. However, it may not be easy for medical companies to obtain patient's personal condition before and after medical treatment. One way to obtain relevant information regarding the effectiveness of drugs are based on online text reviews given by customers.

This paper aims to recognize the effectiveness level of drugs based on informal text in web media, by comparing two models, the Feedforward Multilayer Perceptron (MLP) and Support Vector Machine (SVM). The models are trained by a Drug Review Dataset, with their parameters and hyper-parameters being fine-tuned.

1.2 Multilayer Perceptron (MLP)

Multilayer Perception is a widely used supervised Artificial Neural Network. The model builds up a complex network with input layer, output layer and at least one hidden layers with numbers of neurons within each layer. Input data is passed from input layer through hidden layers to output layer. Data is transformed linearly in each layer and the model handles non-linear problems with a non-linear activation as output of each layer. Model is trained iteratively by updating the weight and bias with a loss function.

MLP could handle complex data problems with their number of hidden layers and neurons, and also handle non-linear problems in reality. In addition, although it is computational expensive to calculate the gradient descent and differential functions, it works easily with paralleled computing. Besides, although the parameter-tuning process in training is long, the training time per model is shorter than that of SVM. It requires efforts in fine-tuning the parameters of networks.

MLP updates weights/ bias and converges based on the local minimum loss. Therefore, the model may stop converging before reaching the global minimum. As initial weights and bias are usually called randomly, more cross validations are required. Lastly, considering the interpretability of network, MLP is likely be a black box to explain the classification task.

1.3 Support Vector Machines (SVM)

Support Vector Machine is a supervised algorithm initiated for binary classification, targeting to create a hyperplane in the N-dimensional space that maximize the distance between data points of two classes and the hyperplane. The model is trained by finding suitable kernel function and adding regularization parameter "C" (which minimizes the impact of noisy data).

SVM performs well in both linear and non-linear problems, by imposing a kernel function (e.g. polynomial, radius basic function) for the classifier, which makes the data linearly separable. In addition, SVM gives remarkable performance on handling high-dimensional feature space since it only concerns on the marginal distance from data points, instead of number of features [2]. This makes it a good option for text classification due to numerous text features and the sparsity of words. Lastly, SVM does not require lots of parameter tuning compared to MLP, hence allows automatic tuning by grid search.

However, the training time for SVM is long, especially when kernel tricks (particularly Radius-based Function) is trained. SVM is also sensitive to noisy data as it treats data points equally. Besides, SVM is a binary classifier, pairwise comparisons are performed (One Vs. Rest approach) to conduct multiclass classification [3].

1.4 Hypothesis Statement

Considering the advantages and drawbacks of two models, it is proposed that SVM yields better performance than MLP in text classification.

2 Dataset and Pre-Processing of Text

2.1 Class Labels

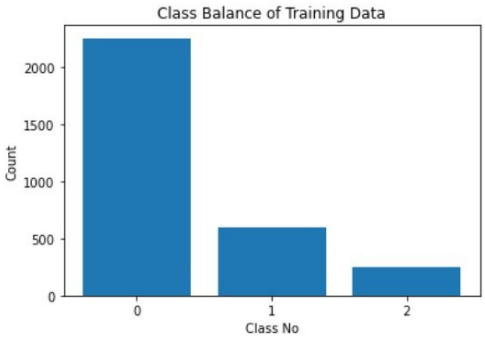

Fig. 1. Class Balance on Effectiveness Ratings in Training Dataset

The data adapted in this paper provides patients' reviews on specific drugs along with related condition, including 3 separate reviews on side-benefit, benefit, and overall comments on the specific drugs. The dataset had been initially pre-split into a training dataset (with 3099 non-null records) and a test dataset (with 1036 non-null records) in a 75:25 ratio.

Besides, the dataset is labelled with 5 effectiveness ratings, which are later categorized into 3 classes ("Highly Effective/ Considerably Effective" [Class 2], "Marginally Effective/ Moderately Effective" [Class 1], "Ineffective" [Class 1]) in this study. These labels are set to ordinal data for further modelling. As in Fig.1, as data imbalance is observed among training data, this may impose negative impacts on positive prediction and accuracy of minor classes (Class 0 & 1). To overcome the issues, over-sampling technique, SMOTE, is adapted to foster class balance.

The data also provides labels for side-effectiveness ratings, yet both side-benefit reviews and ratings are not studied in the paper.

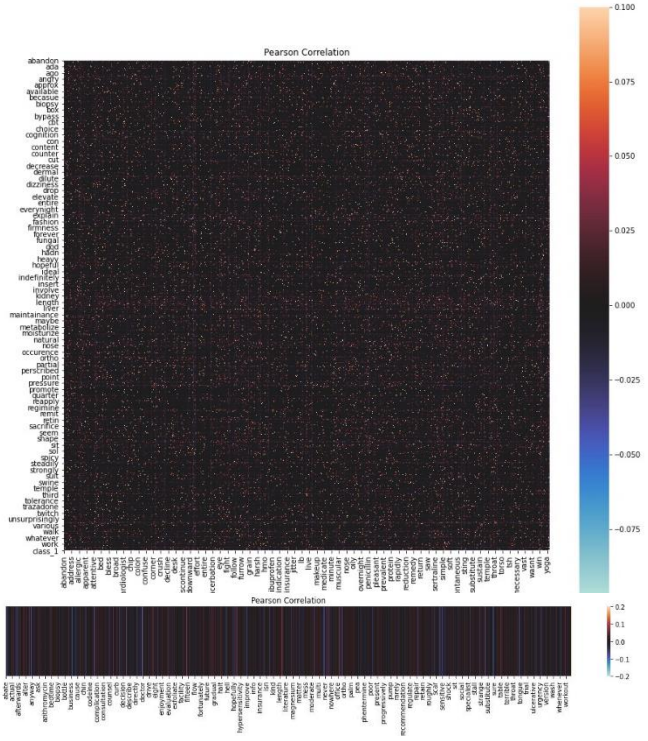

Fig. 2. Pearson Correlation between TF-IDF values of (a) tokenized reviews and (b) labels and tokenized reviews

2.2 Pre-Processing Text

As reviews are unstructured text data, benefit reviews and overall comments are combined and tokenized into single words in 1-gram approach. Since words in web are usually informal with typos, each word for training is expected to exist in at least two records before taking as model features. Common stop words, punctuations, medical condition, and drugs' name are removed from data to ensure data distribution of drugs/ condition before treatment is independent of the classification task. Words are lastly lemmatized to group all tokenized words to their roots. TF-IDF values (continuous) are later calculated for model training by grid-searching the best parameters.

Initial correlation analysis between each single word and between words and labels are found to be not linearly dependent on other features (Fig.2), which suggests suitability of non-linear models. The dataset is also found to be sparsely distributed, with 4636 features/ words. It is observed that the average TF-IDF values of certain words are higher than their counterparts when compare by class.

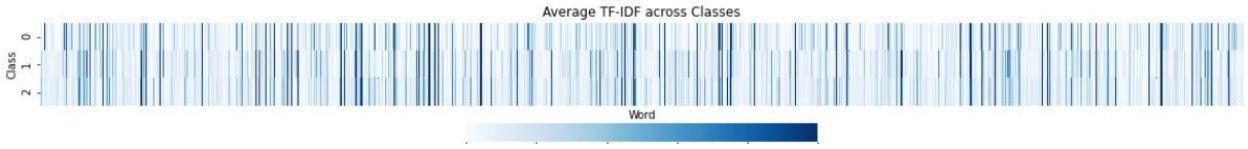

Fig. 3 Average TF-IDF values of Tokenized Words Per Class

3 METHODOLOGY

3.1 Performance Metric

Models are evaluated in the parameter tuning process and in model selection based on several performance metric, including the classification accuracy, F1 score, total time for training and difference between accuracy of training and validation dataset. Since it is a multiclass classification task, F1 score per class is evaluated.

In parameter tuning process, due to the imbalance data, best parameter/ model is selected first by comparing the F1 score per class, followed by generalized accuracy and discrepancy of train-validation accuracy. If similar results are yielded, shorter training time would be considered.

3.2 Validation Training Results

To avoid overfitting and closely monitoring the training process, cross-validation approach is adapted in each parameter choice in model training. Considering the dataset size (3099 records), dataset is cross validated with a comparatively lower value of k=5. Stable accuracy is expected in each fold for good parameter choice/ model selection. Parameter-tuning is decided by averaging the performance metric across k-folds. After the best-performing parameter has finalized, the training data is split into 80:20 dataset to train network weights and bias.

3.3 Architecture of Multilayer Perceptron and Parameter Tuning Approach

A back-propagation approach is adapted to train and update the weights and bias of neurons and hidden layers in every iteration. The network is initialized by some number of hidden neurons and layers at certain learning rate, where random weight and bias are assigned. The computed input values in each layer are then activated by a non-linear function, ReLU, as it converges very quickly and output positive values only. As data is passed to output layer, error with target label is captured by a loss function, Cross-Entropy, which work well to capture information loss for multi-class classification. The local minimum is then passed back to update weights and bias by an optimizer, Adam, which handles the issues of getting stuck in certain local minimum before reaching to global minimum by tuning the learning rate and adding momentum for training.

On top of that, to avoid overfitting, an early stopping is imposed if validation accuracy has reduced for some continuous frequency. Certain weighting may be incurred on the loss function in handling imbalance issue.

The backpropagation process is iterated for 300 training epochs in each model tuning and the parameters are tested with a grid search approach. Hyperparameters are varied to obtain best-performing model, including learning rating, number of hidden layer, number of neurons in hidden layers, early-stopping criteria or other regularization measures. Some parameters for text pre-processing are also tuned to yield best performance, including maximum word feature, minimum document frequency and number of k-neighbour for oversampling in SMOTE.

3.4 Architecture of Support Vector Machine and Parameter Tuning Approach

Two key components of SVM are the suitability of kernel function and the regularization parameters to reduce impacts of noisy data. Kernel function such as Polynomial and Radius-Based Function would be tested with varying hyperparameters, including degree and gamma value. Grid search method for each kernel function is tested.

4 Experimental Results and Parameter choice

	Training	Training	Validation Dataset				Test Dataset			
	Accuracy	Time (s)	Accuracy	F1 Score Class 0	F1 Score Class 1	F1 Score Class 2	Accuracy	F1 Score Class 0	F1 Score Class 1	F1 Score Class 2
MLP	95.89%	65.6	70.96%	0.19	0.25	0.83	68.73%	0.18	0.24	0.82
SVM	85.64%	0.697	74.03%	0.33	0.13	0.86	71.33%	0.35	0.17	0.84

Table 1. Training, validation, and Test dataset Performance Metric

4.1 Pro-processing Text

The best-performing text pre-processing includes the TF-IDF vectorization of reviews to maximum word feature of 3000 (out of 4636 words), and a minimum document frequency of 3. On the other hand, SMOTE oversampling the minority classes (class 0& 1) and majority to ratio of 0.4:0.3:1, whereas at least 5 neighbours are needed for performing SMOTE.

4.2 Multilayer Perceptron

The best model of MLP is constructed by 1 hidden layer with 40 neurons within the layer. A non-linear activation function, ReLU, is used to produce outputs. The loss is calculated by cross-entropy with address to class weight and optimized by Adam at a learning rate of 0.001. An early stopping is implemented within the 300 training epochs, whereas 5 continuous increase of validation loss would stop the training.

4.3 Support Vector Machine

The best model of SVM is constructed by a linear kernel and has added a penalty to the decision boundary (C=0.121) to improve the prediction of minority class. The loss from actual value is calculated by hinge-square function at l2 penalty. No extra class weight is incurred in the algorithm.

5 Analysis and Critical Evaluation of results

5.1 Model Selection upon Performance Metric

Parameters and hyperparameters are decided upon the performance metric of validation as discussed in 3.1-3.2. The decision of MLP parameters often comes with a trade off between F1 score, accuracy, and at expense of overfitting. Accuracy and overfitting solution perform best in 10-20 neurons at rate 0.001 but reported only 0.01/0.1 in F1 score (class 1) (Fig.4). Early stopping criteria at 2 works well with overfitting, yet also reported low F1 score until being stable at 4-5 (Fig 5).

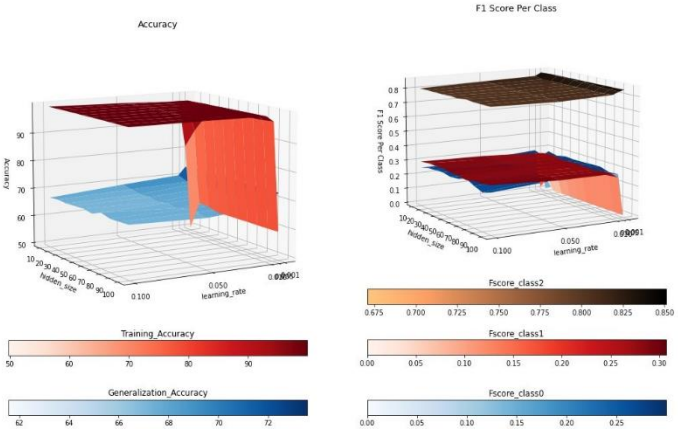

Fig. 4 Grid Search result of Number of Learning rate [0.0001,0.001,0.005,0.01] and number of hidden neurons [10-100, step = 10] on F1 score and accuracy

In addition, MLP is sensitive to imbalance data, as the loss function reflects mostly loss from the majority class, and thus weight and bias tuning are mostly affected by majority class [5], although loss has been weighted (loss per class = loss*class weight of training data). Moreover, the high dimension of textual data increases its sensitivity on data imbalance.

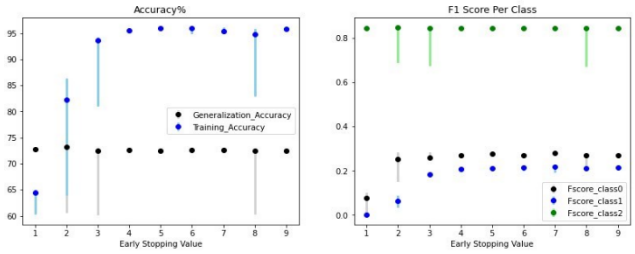

Fig. 5 Variation of performance matric on early stopping criteria, iterated for 3 times.

Model selection of SVM also reported trade off in F score of class 1 with validation accuracy and class 0, 2's F1 score, as observed in Fig.6. Decision is made in favour of class 2, 0 and validation accuracy since it is more important to identity reviews with negative comments/ rated as "highly effective". Besides, overfitting is lower. Linear kernel outperforms RBF kernel at all performance metrics.

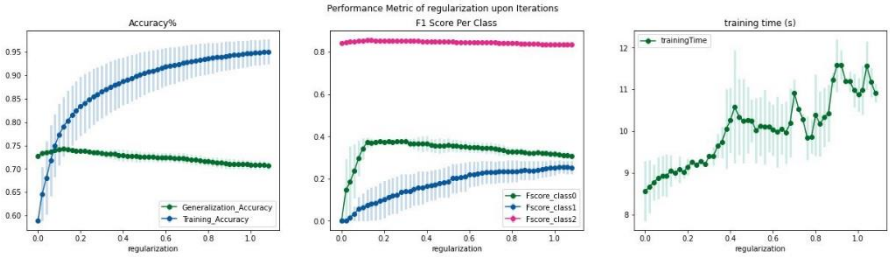

Fig. 6 Performance Metric (Accuracy, F1 Score, Training time) of **Linear Kernel** with regularization "C" at range [0.001 – 1.1] with step = 0.02, tested in both hinge and hinge-square function

5.2 Evaluation and Comparison on Model

Comparing the two algorithms, SVM obtained higher accuracy and suffered less from overfitting. When compare by precision, SVM could make better prediction than MLP in class 2, the majority class, and class 0, one of the minority class, while MLP reported better performance in class 1 prediction. In addition, in terms of training time, SVM, due to the simplicity of linear kernel, spent much less time on training. The architecture in SVM is also less complex than MLP architecture. It outperforms MLP in both model selection stage and in algorithm comparing stage.

Nevertheless, both algorithms poorly perform in classifying minority classes (class 0&1), despite SMOTE is implemented for training data. Both algorithms have observed tendency to overfit, and to reduce its validation accuracy when more attempts are made to correctly classify minority classes, for instance, increasing the regularization measure in SVM, or increasing the neurons in MLP. Insufficient data could be a major reason behind. As a high-dimensional dataset with 3000 input features but with only 3099 for training and majority belongs to class 2, the insufficient data in minority data may fail to capture the variance of real pattern (which SMOTE fail to achieve). As seen in Fig. 7, validation accuracy is highest when word feature is high. However, both class 1 and 0's F1 score are optimal at feature = 2000 (document freq.=4-5), whereas fewer noises are captured.

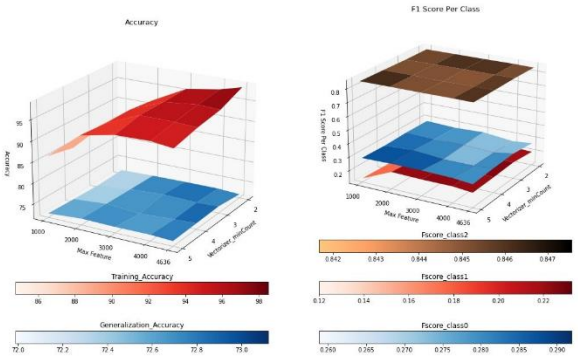

Fig. 7 Grid Search for TF-IDF vectorization parameter (Maximum feature & Minimum document frequency) – Accuracy and F1 score (MLP model)

Practically speaking, it is crucial to identify the real patients who find the medical treatment ineffective (Class 0)/ highly effective (Class 2) in a medical problem from their online reviews. In other words, sensitivity, or say recall is more crucial than precision, particularly for class 0 and class 2. As SVM reported higher recall in class 0 and 2 than MLP (Table 2), it is regarded as a better algorithm for this issue.

	Precision		Recall	
	MLP	SVM	MLP	SVM
Class 0	0.25	0.35	0.15	0.35
Class 1	0.43	0.66	0.16	0.1
Class 2	0.74	0.75	0.92	0.95

Table 2 Precision and Recall of two algorithms in algorithm comparison stage. Algorithms with better performance are highlighted.

6 Conclusion and Future works

This paper reviewed two deep learning algorithms, MLP and SVM in performing multiclass classification for unseen text data. Although both models suffer from insufficient data in the context of high dimensional data, both models perform well in classifying reviews rating drugs as 'Highly effective'. In terms of accuracy and F1 score per class in unseen data, it is concluded that SVM perform better in text classification. In addition, SVM reported higher sensitivity to identify patients who reported medical treatment as ineffective/ highly effective, which makes it a more considerable choice.

To improve the current models' performance and to apply the two algorithms for future text classification, it is advised to study the different text pre-processing techniques that target in dimension reduction or increase the data size. For instance, perform Self-Organizing Maps as a tool of dimension reduction, and input into the MLP model. In addition, hashing method can effectively reduce the high dimensional data with improvement in computing cost. Besides, instead of using cross validation, bootstrapping may work better to generate weighted samples from each class with replacement. The idea is to increase the data for training and to reduce the dimension, such that algorithms find it easier to figure out the data pattern.

7 Reference

[1] Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. 2018. Aspect-Based Sentiment Analysis of Drug Reviews Applying Cross-Domain and Cross-Data Learning. In Proceedings of the 2018 International Conference on Digital Health (DH '18). ACM, New York, NY, USA, 121-125

[2] Joachims T. (1998) Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0026683

[3] Karamizadeh, Sasan & Abdullah, Shahidan & Asl, Mehran & Shayan, Jafar & Rajabi, Mohammad. (2014). Advantage and Drawback of Support Vector Machine Functionality. I4CT 2014 - 1st International Conference on Computer, Communications, and Control Technology, Proceedings. 10.1109/I4CT.2014.6914146.

[4] Korde, Vandana. (2012). Text Classification and Classifiers:A Survey. International Journal of Artificial Intelligence & Applications. 3. 85-99. 10.5121/ijaia.2012.3208.

[5] Johnson, J.M., Khoshgoftaar, T.M. Survey on deep learning with class imbalance. J Big Data 6, 27 (2019). https://doi.org/10.1186/s40537-019-0192-5

8 APPENDIX

8.1 Glossary

Dataset Features and Labelled Class

urlDrugName (categorical): name of drug
condition (categorical): name of condition
benefitsReview (text): patient on benefits
sideEffectsReview (text): patient on side effects
commentsReview (text): overall patient comment
rating (numerical): 10 star patient rating
sideEffects (categorical): 5 step side effect rating
effectiveness (categorical): 5 step effectiveness rating

1) Evaluation Methodology & Performance Metric

Cross-Validation: A resampling procedure used to evaluate machine learning models on a limited data sample by splitting data into k folds, train on k-1 fold and validate in remaining fold for k times.
F1 Score: Covey the balance between Recall and Precision by formula {2*((Precision*Recall)/ (Precision+ Recall))}, where recall is the true positive rate and precision is the predictive power of model.
Hyperparameters: values of model parameters which is critical for building accurate and robust model. They discover the balance between bias, variance and weight to prevent overfitting/ underfitting.
Overfitting: Model that achieves high accuracy in training/ seen data but is not performing well in unseen/ new/ validation data

2) Multilayer Perceptron

Model Definition: A class of feedforward, supervised artificial neural network (ANN), which composed of one of more than one hidden layer, one input layer and one output layer with numbers of neurons within each layer. Each layer is activation by a non-linearly function. Learnt data by back-propagation to calculate the weight and bias of network.
ReLU: (Rectified Linear Unit) activation function, which has derivative function to perform back-propagation

3) Support Vector Machine

Model Definition: A binary/ One Vs. Rest supervised deep learning algorithm that perform classification or regression tasks by creating a separation between groups according to the data pattern
Decision boundary: The hyperplane that separate the two groups
C: Regularization measure that is inversely proportional to regularization power. It adds penalty to the decision boundary to fit data closer to the boundary.
Kernel: A window function that take data as input and transform it to the required form. Common kernel includes Radius-based function, polynomial, and sigmoid function.
Gamma: Kernel coefficient for 'rbf', 'poly' and 'sigmoid'
Degree: Degree of the polynomial kernel function
Hinge/ square_hinge: Loss function used by Linear Kernel, which calculate penalty (loss size) for being away from the decision boundary+1. Square_hinge implied loss size in exponential loss (higher penalty).

4) Text Pre-Processing

Word Vectorization: A general process of turning a collection of text documents into numerical feature vectors
TF-IDF: Stands for "Term Frequency — Inverse Document" Frequency, which summarize how frequent a word appears in a text, and how frequent a word appear across all documents.
Tokenization: Process of transforming a stream of words into individual words

8.2 Implementation of Model Training

Multilayer Perceptron

Grid search approach is adapted in MLP training in phrases of "Minority Classes Oversampling Ratio", "K-neighbor in SMOTE & MLP Layer", "Learning rate & Hidden neurons", "TF-IDF parameters" and "Early stopping criteria". Hyperparameters would be re-trained in earlier phrase after reaching the last phrase. Apart from performance metric, training process would also consider the stability across K-folds and iteration stability in training epochs (Fig.9a). As accuracy of minority class is concerned since 1st phrase, attempts are made to capture minority class data in either a second layer, or by increase of neurons. Nevertheless, grid search in 2nd phrase suggested that k-neighbor of SMOTE merely influences on performance, yet 1 layer capture data better (Fig.8) with a stabler iteration/ accuracy across 5 folds. It is also observed almost all validation loss converges fast even in learning rate 0.001 (Fig 9b), attempts are made to reduce learning rate to 0.0001, yet resulted in either 0 in minority's F1 score, or underfitting issues (i.e. even training data distribution not captured) (Table 3). In addition, use of SoftMax activation reduces validation accuracy to 67%.

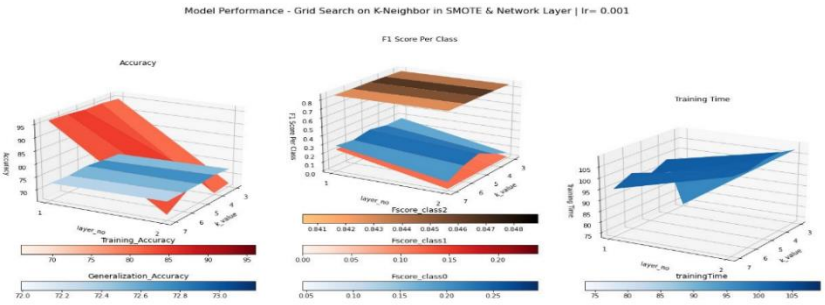

Fig. 8 Grid Search result of Number of Hidden Layer and Number of K-neighbours in Over Sampling Technique (SMOTE)

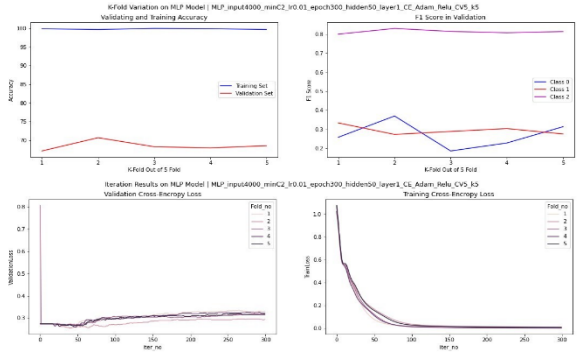

Fig. 9 Grid Search for Minority Class ratio in SMOTE (a) K-Fold variation of (1) Accuracy (2) F1 score per class; (b) Iteration epochs process of (1) Validation loss (2) Training Loss.

LR	Neurons	Val. Accuracy	F1_class0	F1_class1	F1_class2	Overfitting
0.001	20	73.5720	0.2927	0.1049	0.8510	14.000
0.001	30	73.2493	0.2794	0.1882	0.8500	19.661
0.001	40	73.1201	0.2924	0.2186	0.8488	21.192
0.001	10	73.0880	0.2423	0.0161	0.8475	7.880
0.0001	20	72.7654	0.0328	0.0000	0.8427	-12.526
0.0001	30	72.7331	0.0317	0.0000	0.8425	-12.881
0.0001	40	72.7331	0.0000	0.0033	0.8426	-13.986
0.0001	100	72.7009	0.0000	0.0000	0.8421	-13.791
0.0001	50	72.7009	0.0000	0.0000	0.8419	-13.825
0.0001	60	72.7009	0.0000	0.0000	0.8419	-13.840

Table 3 Top 10 best performing model in Grid Search of learning rate (LR) & Neurons number. Yellow indicated the chosen hyperparameter.

Kernel	Kernel parameter	Balanced	Val. Accuracy	F1_class0	F1_class1	F1_class2	Overfitting
Linear	l2; hinge	N	0.72	0.33	0.22	0.84	0.2
Linear	l2;hinge2	N	0.72	0.33	0.22	0.84	0.2
Linear	l2;hinge2	Y	0.75	0.22	0.21	0.86	0.24
Linear	l1; hinge2	N	0.7	0.33	0.31	0.83	0.26
Linear	l1; hinge2	Y	0.66	0.32	0.36	0.8	0.31
Linear	l2 ;hinge2	N	0.7	0.3	0.28	0.83	0.27
Poly	degree = 3	N	0.73	0.04	0.05	0.84	0.23
Poly	degree = 3	Y	0.73	0.06	0.05	0.85	0.24
rbf	gamma = 'scale'	N	0.74	0.19	0.11	0.85	0.24
rbf	gamma = 'scale'	Y	0.75	0.22	0.21	0.86	0.24

Table 4 Baseline model for 3 Kernel type at C=1, with good performing models highlighted in yellow. Also show class weight in SVM algorithm is set as 'balanced'.

Support Vector Machine

Three types of kernel are first trained as baseline model (C=1), while linear kernel, followed by RBF, outperformed (Table 4). Grid search on "C Vs. specific kernel parameter" are conducted. Considering the high dimensionality and sparsity of data, RBF kernel is initially trained from extremely low value 1*10^-100 – 10^-65, C in range([1-11 with step=2]) which results in underfitting (0.59 training accuracy, F1 score of minority class=0, 0.73 val. accuracy).

Both linear and RBF kernel has been tested with ranges of regularization values C. The optimal C of RBF is 2.1, which reported accuracy of 74.4%, F1 score 0.22, 0.18 and 0.85 in 3 classes. Considering overfitting is around 25% and lower F1 score than linear kernel, it is not chosen.

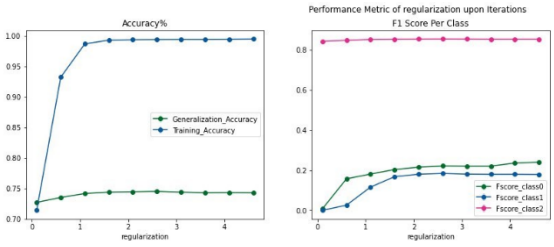

Fig. 10 RBF Kernel in search for optimal regularization measure C, ranging from 0.1 to 5 with step =0.5