CSE2042
MACHINE LEARNING AND ITS APPLICATIONS
J Component Report

A project report titled

Relationships Between Imbalance and Overlapping of Datasets

By

19BPS1006	Vikrant Thoidingjsm
19BPS1036	Amrit Sen
19BPS1081	Anjani Babu Janyavula
19BPS1066	Hanuman Sai

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted to

Dr. Shivani Gupta
School of Computer Science and Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

ABSTRACT

Several works point out class imbalance as an obstacle to applying machine learning algorithms to real world domains. However, in some cases, learning algorithms perform well on several imbalanced domains. Thus, it does not seem fair to directly correlate class imbalance to the loss of performance of learning algorithms. In this work, we develop a systematic study aiming to question whether class imbalances are truly to blame for the loss of performance of learning systems or whether the class imbalances are not a problem by themselves. Our experiments suggest that the problem is not directly caused by class imbalances, but is also related to the degree of overlapping among the classes.

So, we are going to find a solution for the problem of an imbalanced dataset with a problem of overlapping with several algorithms.

INTRODUCTION

Present world is filled with data. We get data from various sources, various fields and domains which have different constraints and situations. As machine learning algorithms are improving day by day to tackle various problems, a variety of challenges arise due to the nature of data. One of the well-recognized challenges that draw attention is class imbalance problem. This problem arises when we have the majority of proportions in a single class. On the contrary, other classes are having less proportion of data. This kind of problem occurs in various real life domains like cancer prediction, credit card fraud prediction, fake email classification etc. where occurrence of other classes have been rare so we don't get a complete proper balanced dataset. So it is one of the most common problems in the machine learning domain.

So while handling these kinds of datasets when we create models, those models will be biased towards the majority class, which gives high accuracy sometimes, which makes the problem harder to solve.

In the time of making progress in this area, the new challenges that researchers are facing, that makes classification even harder, is class overlapping. This situation mostly occurs in the domain of 1) drug design 2) character recognition where data samples from different classes have similar characteristics. There will be minor differences in some aspects which makes it harder to classify.

It is shown that when classes are separated, regardless of imbalance ratio, instances can be classified correctly using standard algorithms. However this class overlapping leads to misclassification. As a result, this issue to resolve is growing algorithmically.

In this paper, we will mainly deal with overlapping in imbalance data, by taking 9 cases of datasets using different combinations.

The aim of the paper is to analyze the relation between class imbalance and class overlapping and the effect of them in performance. We will use different combinations of datasets and analyse the performance of having various factors taken into consideration.

We are also going to analyze real world datasets like credit card and diabetes datasets, which are having similar kind of problems and analyse the performance for them by building various models like - Decision Tree Classifier, KNN Classifier, SVM Classifier, Logistic Regression and Random Forest Classifier, and analyse the performance of algorithms.. We also try to improve the performance for the models by using the SMOTETomek resampling algorithm.

LITERATURE REVIEW

In many of the papers referred, there has been either an introduction of models or improvements of uncommon models which aims to improve the accuracy of the classification of unbalanced or overlapped data and the withstanding relation between these two topics in different models using different algorithms. Improvement of existing algorithms has also been done out of which few had a significant improvement while working with the models

Some of the introductory methods which had few module(s) common as with our model would be the introduction of a certain approach based on K-nearest neighbour(KNN) proposed by Tang in his paper "*Improved classification for problem involving overlapping patterns*"[1] which aims to extract the vague regions in the given data.

Another module which is the inclusion of SMOTE and TOMEK links has been derived as a combination of the concepts introduced by Tomek in his paper " *Two modifications of CNN*"[2] and N.V. Chawla in his paper "*Smote: synthetic minority over-sampling technique*".[3] The efficiency could be verified for both papers as Tomek proves in his paper, the superiority of both modifications/models to the CNN method with its reasons relating to the size and boundary of the design set.

This model has been proven to be efficient in combination in few researches, one of them by Batista et al in his paper "*Balancing strategies and class overlapping*"[4]where he did the comparison of five models in which the SMOTE + TOMEK and SMOTE + ENN Seem to be the most efficient and suitable for the process.

Through the research in this field, there has been a development of a possibility of a relation between the class imbalance and the overlapping of the datasets. Denil showed in his paper ". *Overlap versus imbalance",*[5] the possibility of direct interdependence between the overlapping and the imbalance but was unable to prove it but proved the seriousness of the overlapping issue compared to the imbalance in a dataset.

Even with this result, tackling imbalance still remains an important work for better accuracy. This has been tried by Guillaume in his paper "*Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning*"[6] where he uses over ten different algorithms/methods to achieve this including SMOTE +ENN.

There hasn't been much research on analysing or solving the overlapping methods. A mentionable progress is made by Xiong in his paper, "*Classification with class overlapping: a systematic study*"[7]. A dataset for working with overlapping and imbalance together has been formulated and presented by Almutairi.[8]

Another mentionable work in this field is by V.Garcia in his paper "*An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics*"[9] where the effect of imbalance and overlap on each other has been studied which did not have an accurate resultant relation but there was a certain progression in proving their relation.

EXPERIMENTAL METHODOLOGY

Case Classification using different kinds of datasets for the given problem:

Real-life datasets are always imbalanced and overlapping, and the best way to deal with them is to take both imbalance and overlapping into account at the same time, treating them as a single problem.

In the study conducted by us, with reference to the given research paper, at first the challenges posed by imbalanced and overlapping datasets are scrutinized using synthetic datasets, just as done by the researchers originally, and the obtained results are compared with theirs.

As a further study, we apply the same analysis techniques/algorithms on two real-life datasets, as we did for the synthetic ones. In addition to those techniques, we also apply two other algorithms on these real-life datasets as well.

Understanding the Synthetic Datasets:

As far as the analysis with the synthetic datasets is concerned, there are nine such datasets, of four feature variables each (X1, X2, X3, X4) and a target variable (Y) with two classes (0 and 1), in total, with different combinations of degree of imbalance and overlapping. The description of the datasets with different levels/degrees of class imbalance and feature overlapping/separation are listed as follows:

1. **Unbalanced class with none of the features overlapping.**

2. **Unbalanced class with one feature overlapping.**

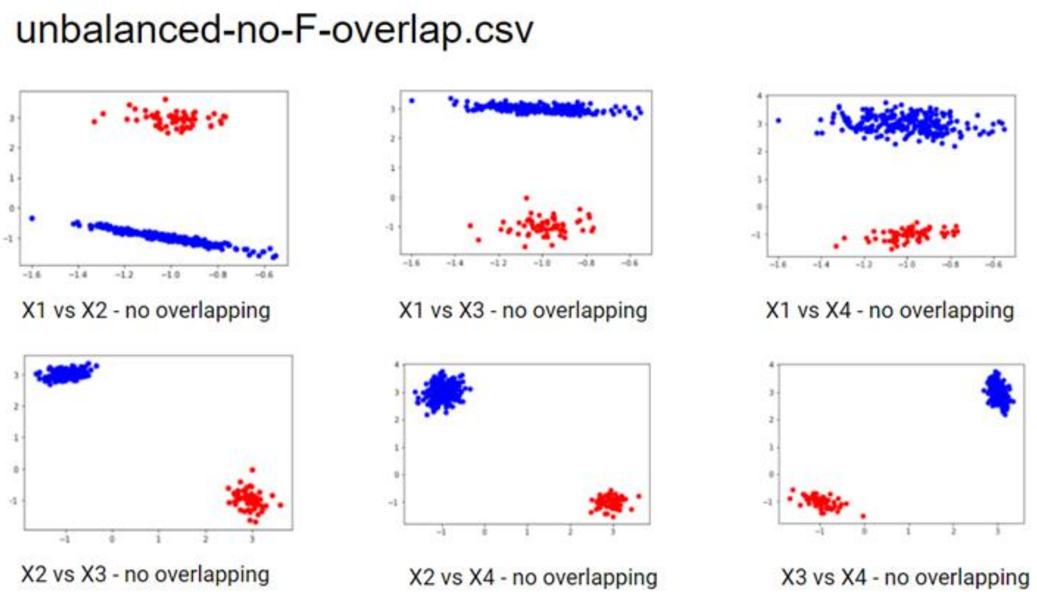
3. **Unbalanced class with three features overlapping.**
4. **Unbalanced class with all the features overlapping.**
5. **Balanced class with none of the features overlapping.**
6. **Balanced class with one feature overlapping.**
7. **Balanced class with two features overlapping.**
8. **Balanced class with three features overlapping.**
9. **Balanced class with all of the features overlapping.**

The above datasets have 300 instances each, and are taken from [8], and can be found in detailed form at the same location. The unbalanced datasets have 240 instances corresponding to class 1, and 60 instances corresponding to class 0, indicating significant imbalance. Whereas, the balanced datasets have equal share of class 1 and class 0 instances i.e.150 instances each.

The above datasets are respectively named in the CSV format as follows:

1. **unbalanced-no-F-overlap.csv**
2. **unbalanced-one-F-overlap.csv**
3. **unbalanced-3-F-overlaple.csv**
4. **unbalanced-all-F-overlap.csv**
5. **Balanced-no-F-overlap.csv**
6. **Balanced-one-F-overlap.csv**
7. **Balanced-2-F-overlap.csv**
8. **Balanced-3-F-overlap.csv**
9. **Balanced-All-F-overlap.csv**

A graphical visualization of the levels of overlapping and separation, for the different types of imbalanced and balanced datasets is gives follows:



X1 vs X2 - no overlapping

X1 vs X3 - no overlapping

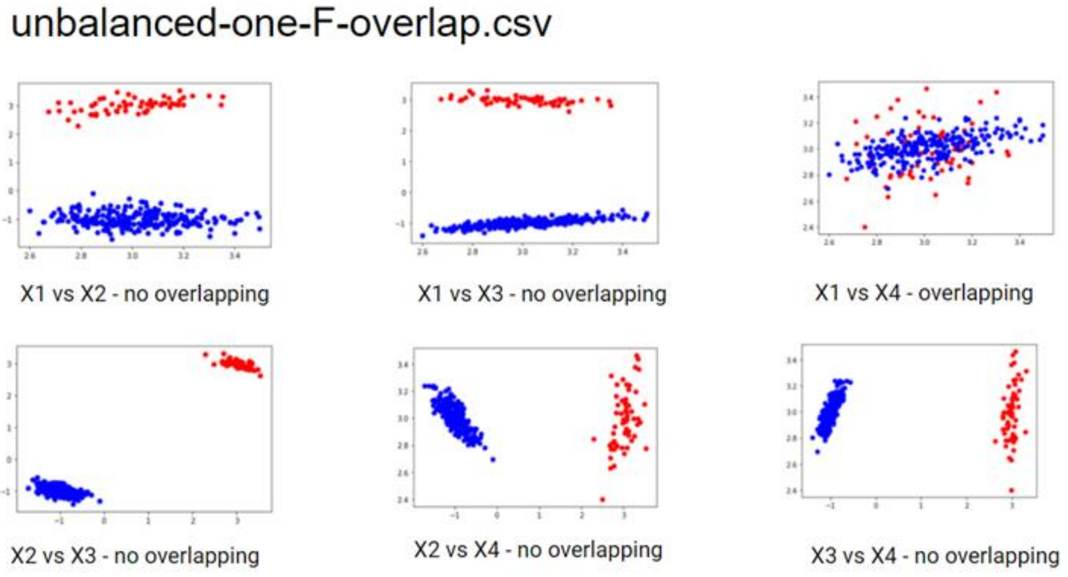
X1 vs X4 - no overlapping

X2 vs X3 - no overlapping

X2 vs X4 - no overlapping

X3 vs X4 - no overlapping

Fig.1 – The plot for one feature vs another, covering all 6 combinations of the four feature variables of the *unbalanced dataset with none of the features overlapping*. Here, as already shown, none of the features overlap, and the level of separation between each class, for all the non-overlapping features, is also very high.



X1 vs X2 - no overlapping

X1 vs X3 - no overlapping

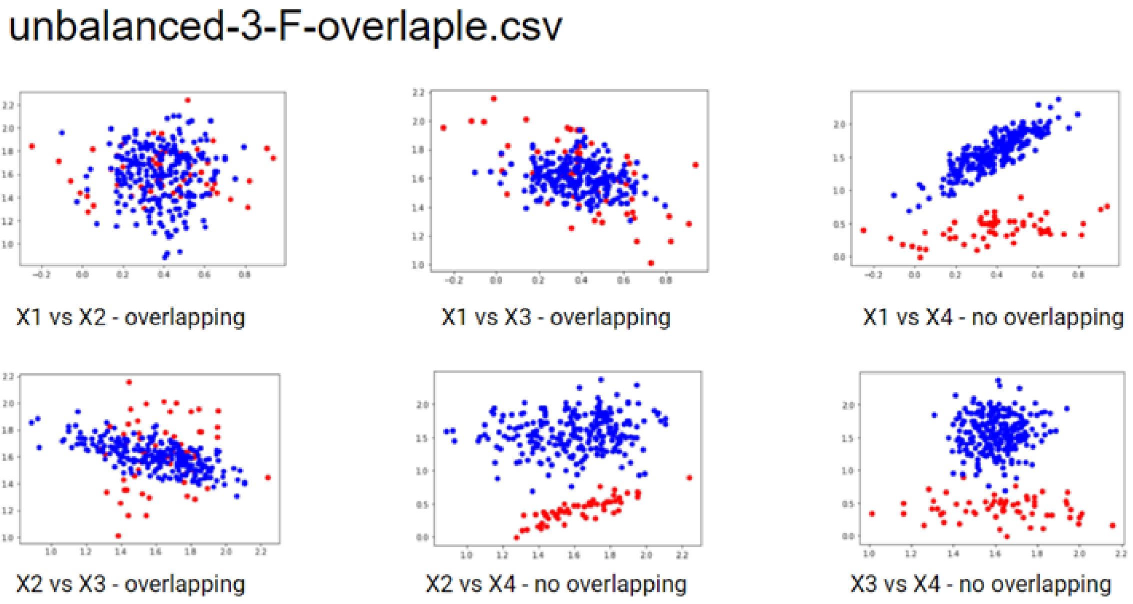
X1 vs X4 - overlapping

X2 vs X3 - no overlapping

X2 vs X4 - no overlapping

X3 vs X4 - no overlapping

Fig.2 - The plot for one feature vs another, covering all 6 combinations of the four feature variables of the *unbalanced dataset with one of the features overlapping*. Here, as already shown, one of the features overlap (plot for X1 vs X4), and the level of separation between each class, for the non-overlapping features, is very high.



X1 vs X2 - overlapping

X1 vs X3 - overlapping

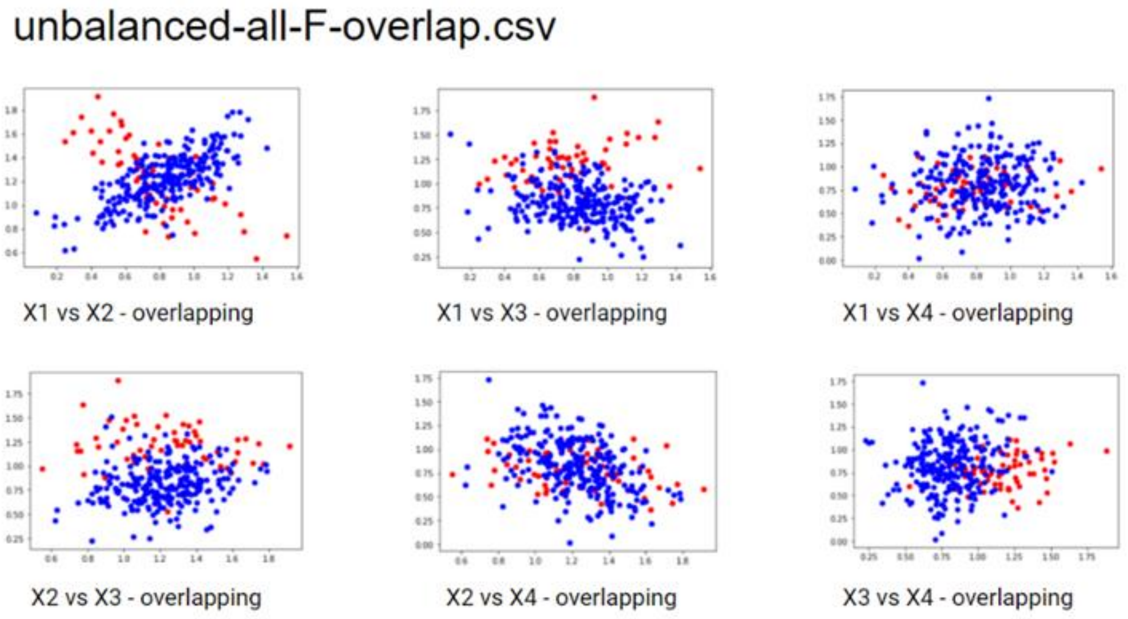
X1 vs X4 - no overlapping

X2 vs X3 - overlapping

X2 vs X4 - no overlapping

X3 vs X4 - no overlapping

Fig.3 - The plot for one feature vs another, covering all 6 combinations of the four feature variables of the *unbalanced dataset with three of the features overlapping*. Here, as already shown, three of the features overlap (the three overlapping plots), and the level of separation between each class, for the non-overlapping features, is very low.



X1 vs X2 - overlapping

X1 vs X3 - overlapping

X1 vs X4 - overlapping

X2 vs X3 - overlapping

X2 vs X4 - overlapping

X3 vs X4 - overlapping

Fig.4 - The plot for one feature vs another, covering all 6 combinations of the four feature variables of the *unbalanced dataset with all of the features overlapping*. Here, as already evident, all the features overlap with each other.

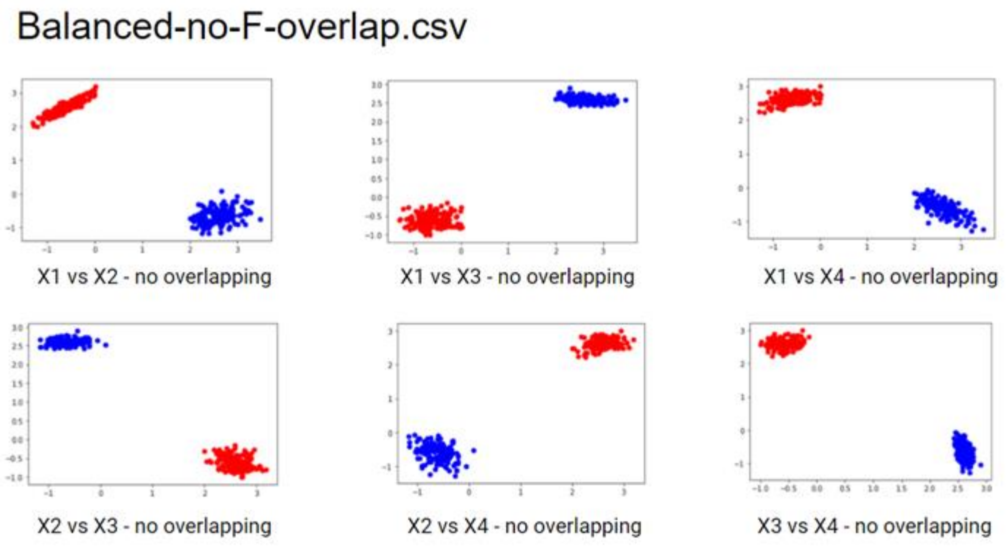


Fig.5 - The plot for one feature vs another, covering all 6 combinations of the four feature variables of the *balanced dataset with none of the features overlapping*. Here, as already shown, none of the features overlap, and the level of separation between each class, for all the non-overlapping features, is very high.

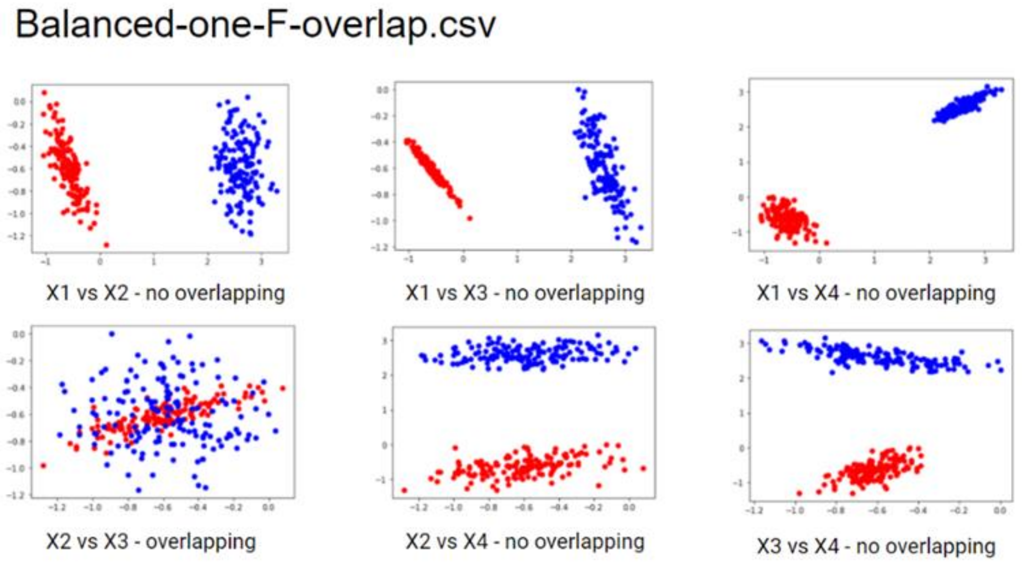


Fig.6 – The plot for one feature vs another, covering all 6 combinations of the four feature variables of the *balanced dataset with one of the features overlapping*. Here, as already shown, one of the features overlap (plot for X2 vs X3), and the level of separation between each class, for the non-overlapping features, is very high.

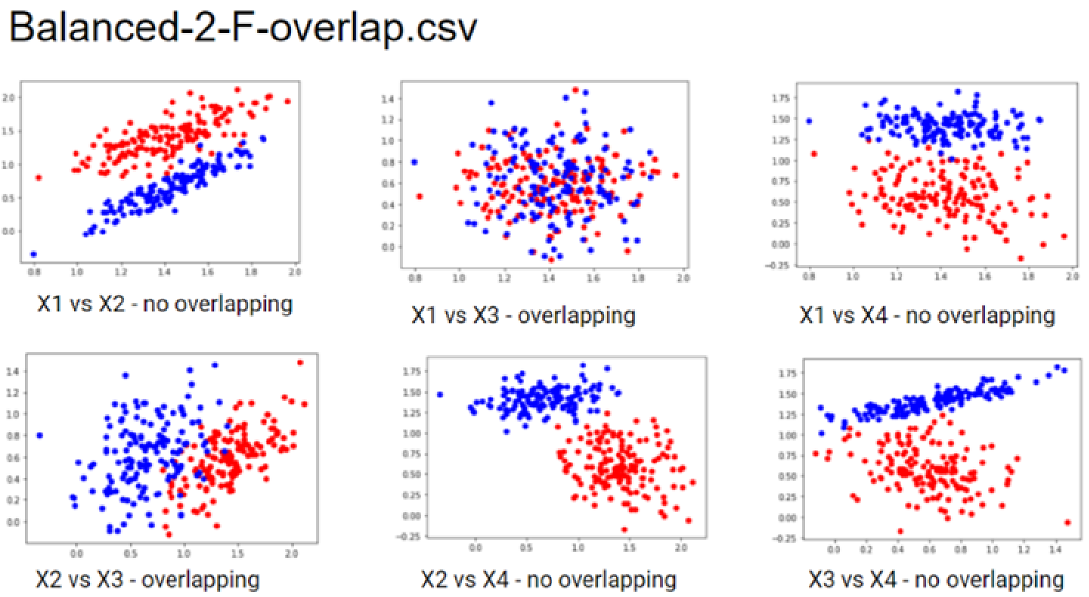


Fig.7 - The plot for one feature vs another, covering all 6 combinations of the four feature variables of the *balanced dataset with two of the features overlapping*. Here, as already shown, two of the features overlap (the two overlapping plots), and the level of separation between each class, for the non-overlapping features, is very low.

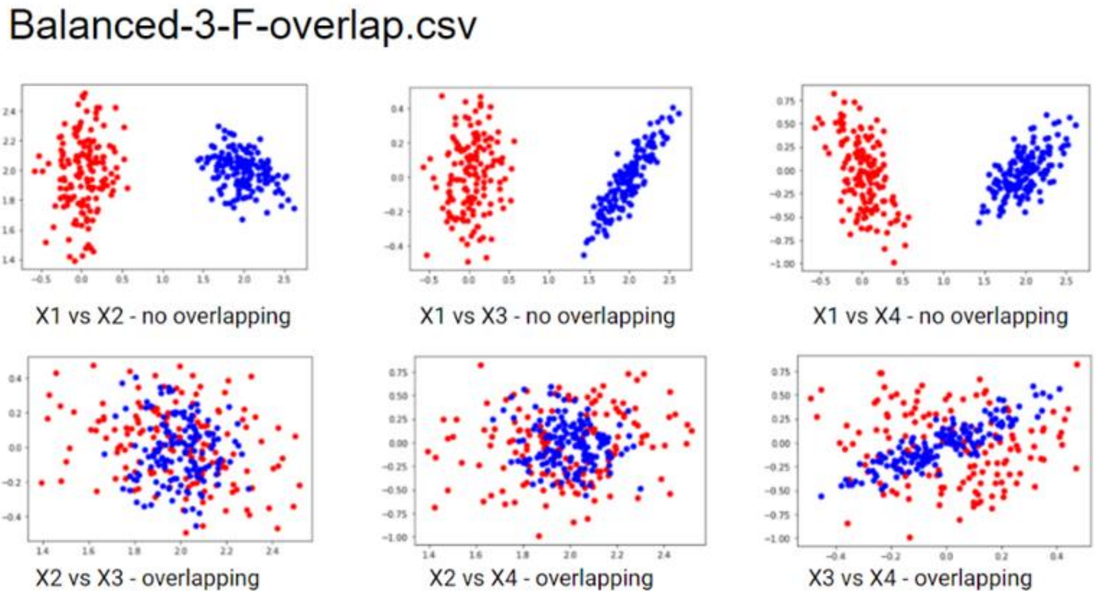


Fig.8 - The plot for one feature vs another, covering all 6 combinations of the four feature variables of the *balanced dataset with three of the features overlapping*. Here, as already shown, three of the features overlap (the three overlapping plots), and the level of separation between each class, for the non-overlapping features, is very low.

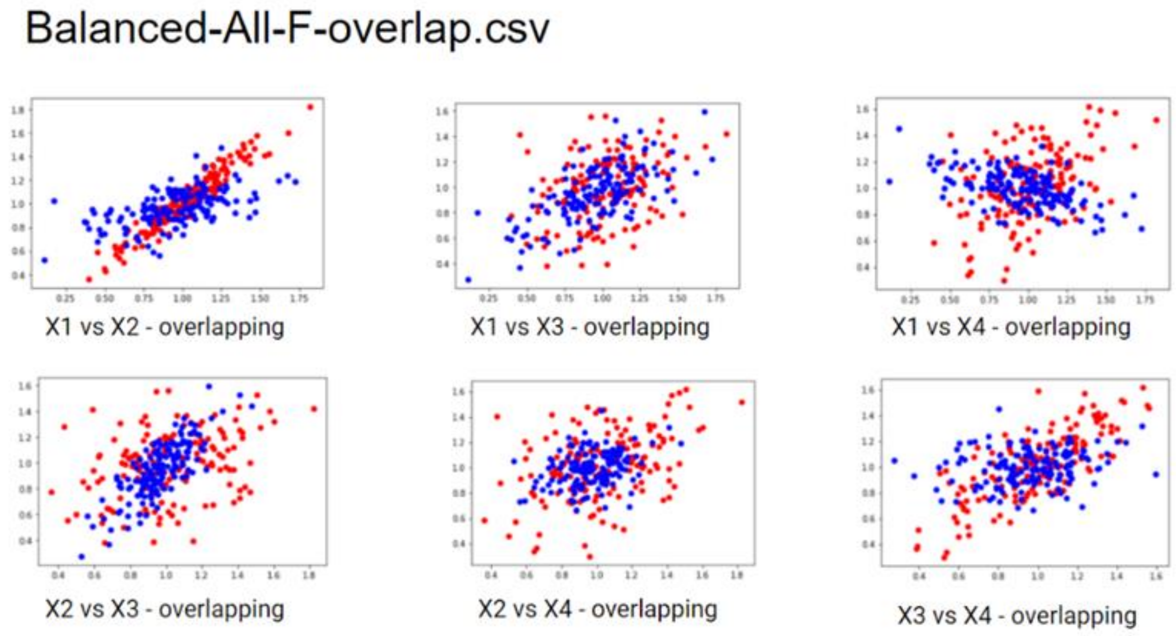


Fig.9 - The plot for one feature vs another, covering all 6 combinations of the four feature variables of the *balanced dataset with all of the features overlapping*. Here, as already evident, all the features overlap with each other.

Here, two types of datasets are used. The first four are the different types of *unbalanced datasets*, with different levels of overlapping over the four features for the two classes. The last five datasets are *balanced datasets*, with different levels of overlapping over the four features for both the classes. In both the types of datasets, the variation in overlapping ranges from 'none of the features overlapping' to 'all the features overlapping'.

Analysis of Unbalanced and Balanced Data with different levels of Overlapping:

At first, we perform the Statistical Consistency Analysis for all the nine datasets, with respect to the two classes (0 and 1) present in the target variable, and understand the differences which are obtained. For ease of understanding, the results obtained by our model, for the *unbalanced and balanced dataset with all features overlapping*, are tabularized as follows:

Table 1 – Statistical Consistency Analysis for the *unbalanced dataset with all features overlapping.*

Unbalanced-all-F-overlap	Class	Feature 1 (X1)	Feature 2 (X2)	Feature 3 (X3)	Feature 4 (X4)
N (Number of instances)	0	60	60	60	60
	1	240	240	240	240
Mean	0	0.7735	1.2335	1.1907	0.7709
	1	0.8302	1.2188	0.8088	0.8244
Variance	0	0.068961	0.088261	0.048722	0.031709
	1	0.053322	0.043014	0.042207	0.076211
Skewness	0	0.50345441	0.0098331	0.12549064	-0.05878412
	1	-0.32946883	-0.03762776	0.24914657	0.01454974

Table 2 - Statistical Consistency Analysis for the *balanced dataset with all features overlapping.*

Balanced-all-F-overlap	Class	Feature 1 (X1)	Feature 2 (X2)	Feature 3 (X3)	Feature 4 (X4)
N (Number of instances)	0	150	150	150	150
	1	150	150	150	150
Mean	0	1.0140	1.0245	0.9934	1.0220

	1	0.9544	0.9696	0.9583	0.9976
Variance	0	0.065321	0.070223	0.065216	0.072375
	1	0.077903	0.023494	0.047822	0.018762
Skewness	0	0.15162464	0.09322401	-0.21753477	-0.26294601
	1	-0.20842954	-0.02883173	-0.14427437	0.01866206

The above Statistical Consistency Analysis of the *unbalance* and *balanced datasets*, using some standard statistical measures like mean, variance, and skewness demonstrates that the presence of a significant skewness value in the overlapping features, renders it difficult to train the model due to the presence of the mixed boundary between the classes. Similar results are obtained by the researchers as well.

As we proceed further, the main motive of our study becomes the quantitative analysis of the nine given datasets using three performance indicators, namely ‘Accuracy’, ‘Precision, and ‘Recall’, and comparing the values obtained by our model with those obtained by the researchers, when the datasets are trained and tested for the same classification algorithms.

To understand the three above mentioned performance indicators (‘Accuracy’, ‘Precision, and ‘Recall’), one must have a basic understating of the ‘Confusion matrix’, which is a matrix showing the complete results of correctly and incorrectly classified data points for each class, and looks like as given below:

		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FP	TP

The ‘Confusion Matrix’ consists of four important values namely ‘TN’, ‘FP’, ‘FN’, and ‘TP’, where:

- **TN** – **True Negative** is an outcome where the model correctly predicts the negative class.
- **FP** – **False Positive** is an outcome where the model incorrectly predicts the positive class.
- **FN** – **False Negative** is an outcome where the model incorrectly predicts the negative class.
- **TP** – **True Positive** is an outcome where the model correctly predicts the positive class.

The *Confusion Matrix* is essential in the context of performance indicators, as the values of *TN, FP, FN,* and *TP* are finally used to calculate the values of *Accuracy, Precision,* and *Recall* as shown below:

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

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Training the Data for the Nine given Synthetic Datasets, and Subjecting it to Different Algorithms for Analysis:

Each of the datasets are split into 75% training data and 25% testing data, and trained using 10-folds cross validation, before subjecting to different machine learning algorithms, namely *Decision Tree Classifier, K-Nearest Neighbour (KNN) Classifier*, and *Support Vector Machine (SVM) Classifier*. The performance indicator (*Accuracy, Precision, Recall*) values obtained for our model are then tabularised, and compared to those obtained by the researchers, for each of the three mentioned machine learning algorithms.

Decision Tree Classifier:

The following table lists and compares the performance indicator (*Accuracy, Precision,* and *Recall*) values obtained by our model, with those obtained by the researchers, on applying the *Decision Tree Classifier* on the nine given synthetic datasets:

Table 3 – Tabularized comparison between the performance indicators obtained for our model with respect to the researcher's model, when *Decision Tree Classifier* is applied.

Decision Tree Classifier	**Accuracy%**		**Precision%**		**Recall%**	
	Paper's	**Our's**	**Paper's**	**Our's**	**Paper's**	**Our's**
Balanced-2-F-overlap	98	99.130	98.124	98.333	97.881	98.182
Balanced-3-F-overlap	100	100	100	100	100	100
Balanced-no-F-overlap	100	100	100	100	100	100
Balanced-one-F-overlap	100	100	100	100	100	100
Balanced-all-F-overlap	70.667	68.458	70.150	72.889	70.382	68.636
Unbalanced-3-F-overlap	98	98.182	97.213	98.947	97.016	98.889
Unbalanced-no-F-overlap	100	100	100	100	100	100

Unbalanced-one-F-overlap	100	100	100	100	100	100
Unbalanced-all-F-overlap	87.333	87.134	79.273	91.304	79.742	91.111

K-Nearest Neighbour (KNN) Classifier:

The following table lists and compares the performance indicator (*Accuracy*, *Precision*, and *Recall*) values obtained by our model, with those obtained by the researchers, on applying the *KNN Classifier*, with K=3, on the nine given synthetic datasets:

Table 4 - Tabularized comparison between the performance indicators obtained for our model with respect to the researcher's model, when *KNN Classifier* is applied.

KNN Classifier	**Accuracy%**		**Precision%**		**Recall%**	
	Paper's	**Our's**	**Paper's**	**Our's**	**Paper's**	**Our's**
Balanced-2-F-overlap	99	100	99.062	100	98.941	100
Balanced-3-F-overlap	100	100	100	100	100	100
Balanced-no-F-overlap	100	100	100	100	100	100
Balanced-one-F-overlap	100	100	100	100	100	100
Balanced-all-F-overlap	76.167	79.644	76.358	76.041	75.815	84.697

Unbalanced-3-F-overlap	98.833	100	98.514	100	97.883	100
Unbalanced-no-F-overlap	100	100	100	100	100	100
Unbalanced-one-F-overlap	100	100	100	100	100	100
Unbalanced-all-F-overlap	90.667	96.443	86.495	96.833	83.785	98.856

Support Vector Machine (SVM) Classifier:

The following table lists and compares the performance indicator (*Accuracy, Precision*, and *Recall*) values obtained by our model, with those obtained by the researchers, on applying the *SVM Classifier* on the nine given synthetic datasets:

Table 5 - Tabularized comparison between the performance indicators obtained for our model with respect to the researcher's model, when *SVM Classifier* is applied.

SVM Classifier	**Accuracy%**		**Precision%**		**Recall%**	
	Paper's	**Our's**	**Paper's**	**Our's**	**Paper's**	**Our's**
Balanced-2-F-overlap	99.333	100	99.375	100	99.294	100
Balanced-3-F-overlap	100	100	100	100	100	100
Balanced-no-F-overlap	100	100	100	100	100	100

Balanced-one-F-overlap	100	100	100	100	100	100
Balanced-all-F-overlap	68.333	77.727	69.925	72.032	68.511	90.985
Unbalanced-3-F-overlap	99.222	100	99.009	100	98.589	100
Unbalanced-no-F-overlap	100	100	100	100	100	100
Unbalanced-one-F-overlap	100	100	100	100	100	100
Unbalanced-all-F-overlap	89	95.099	85.323	95.228	78.344	98.889

Inference drawn from the above Classifiers (applied without resampling) for the synthetic datasets:

In the above implementations, three machine learning algorithms (*Decision Tree, KNN, SVM Classifier*) are applied on *four unbalanced datasets*, and *five balanced datasets*, all of which have different levels of overlapping and separation. The first five datasets in the above three tables correspond to the balanced datasets and the last four datasets correspond to the unbalanced datasets, where the 'F' in their name stands for feature. The obtained performance indicators are compared with those obtained by the researchers. In most cases, the results obtained by us are similar to those obtained by the researchers, however, when different values are registered, it is found that our model obtains improved results.

Here, it is found that all those datasets with small overlapping with significant separation among the features, are correctly classified by all three classifiers. In other words, on precisely observing, we find that for our model, the three algorithms render *100% accuracy, precision* and *recall* values for the *balanced datasets with zero, one and three features overlapping*, and the *unbalanced datasets with zero and one feature overlapping*. This is the case for both our model and the researcher's model. Our model also obtains *100% values for* *the three performance indicators*, for the *balanced dataset with two features overlapping* and the *unbalanced dataset with three features overlapping*, on applying the above classifiers, except for the *Decision Tree Classifier*. However, this is not the case for the results obtained by the researchers, whose corresponding values for these two datasets are less than 100%. Even for those datasets, where the values obtained are less than 100%, our model obtains improved values with respect to those obtained by the researchers, as is evident from the above tables.

Another observable inference is that, although the results obtained by the KNN and the SVM classifiers for the *balanced dataset with one and three features overlapping* are the same (*100% accuracy, precision, recall*), for the *Decision Tree Classifier,* better results are obtained for the *balanced dataset with three features overlapping* as compared to the one with *two features overlapping*. The reason behind this can be attributed to the fact that non-overlapping features have more separation between them for the *balanced dataset with three features overlapping* (Fig.8), as compared to the *balanced dataset with two features overlapping* (Fig.7). The same reason can be extended for the similar results obtained by the researchers.

We can say that for all the balanced and unbalanced datasets, all the three classifiers have performed extremely well, when the overlapping between the features is small with a significant amount of separation between them. This is the case for both our model and the researcher's model.

Another expected observation is that, when all the features in a dataset overlap, we again obtain similar, but worse performance indicator values for all the three classifiers. However, the results for this case are better for an *unbalanced dataset with all features overlapping* as compared to the *balanced dataset with all features overlapping*. The reason behind this can be explained with the fact that unbalanced datasets consist of less overlapping between features (less data in the overlapping region), as compared to the balanced datasets.

Resampling the data for the Unbalanced Datasets:

For unbalanced datasets, resampling can be performed for improving the values of *Accuracy, Precision,* and *Recall.*

In the given research paper, the researchers perform oversampling using *ADASYN Algorithm*, and undersampling using *Random Under Sampler Algorithm*, and train the model for the *unbalanced dataset with all features overlapping*, and found the three performance indicator values for the same dataset, using the same set of classification algorithms, as done before.

However, in our model, we used the SMOTETomek algorithm for resampling the data, with the application of the oversampling (using SMOTE) and under-sampling (using Tomek links) technique, and found the three performance indicator values for all the unbalanced datasets, using the same set of classification algorithms (*Decision Tree, Knn, SVM*), like the researchers. SMOTETomek is a technique, which lies somewhere between oversampling and under-sampling, integrating both types of resampling together. SMOTETomek is a hybrid mixture of the oversampling and undersampling methods, and it uses an under-sampling method, called Tomek, alongside an oversampling method, called SMOTE.

The three performance indicators as obtained by our model, after the application of SMOTETomek on the same set of classification algorithms (*Decision Tree, Knn, SVM*) is as given in the table below:

Decision Tree Classifier:

Table 6 - Tabularized data to show the performance indicators obtained for our model, when *Decision Tree Classifier* is applied after resampling all the four *unbalanced datasets* using *SMOTETomek.*

Decision Tree Classifier	**Accuracy%**	**Precision%**	**Recall%**
Unbalanced-3-F-overlap	98.729	99.474	98.889
Unbalanced-no-F-overlap	100	100	100
Unbalanced-one-F-overlap	100	100	100
Unbalanced-all-F-overlap	89.667	92.913	88.627

K-Nearest Neighbour (KNN) Classifier:

Table 7 - Tabularized data to show the performance indicators obtained for our model, when *KNN Classifier* is applied after resampling all the four *unbalanced datasets* using *SMOTETomek*.

KNN Classifier	**Accuracy%**	**Precision%**	**Recall%**
Unbalanced-3-F-overlap	100	100	100
Unbalanced-no-F-overlap	100	100	100
Unbalanced-one-F-overlap	100	100	100
Unbalanced-all-F-overlap	97.731	98.392	97.712

Support Vector Machine (SVM) Classifier:

Table 8 - Tabularized data to show the performance indicators obtained for our model, when *SVM Classifier* is applied after resampling all the four *unbalanced datasets* using *SMOTETomek*.

SVM Classifier	**Accuracy%**	**Precision%**	**Recall%**
Unbalanced-3-F-overlap	100	100	100

Unbalanced-no-F-overlap	100	100	100
Unbalanced-one-F-overlap	100	100	100
Unbalanced-all-F-overlap	96.129	97.248	96.046

Inference drawn from the above Classifiers (applied after resampling) for the synthetic datasets:

After applying the same three classification algorithms on the nine synthetic datasets, after resampling them using SMOTETomek, we find that, for the the *unbalanced datasets with zero and one feature overlapping*, the performance indicator values remain the same i.e., 100% for all the three performance indicators (as this value cannot be improved further). Same is the case for the *unbalanced dataset with three features overlapping*, when the *KNN* and the *SVM* classifiers are applied on it.

However, after resampling, the performance indicator values for the *unbalanced dataset with three features overlapping* improves and reaches 100%, on the application of the *Decision Tree Classifier*. This was not the case without resampling, thus suggesting that resampling indeed improved the performance indicators.

A similar improvement in the performance indicator values is found for the *unbalanced dataset with all features overlapping*, after applying SMOTETomek for resampling. Although the values do not reach 100% for any of the classifiers, a significant improvement in the accuracy, precision and recall percentages is observed, thus proving the credibility of the SMOTETomek resampling technique.

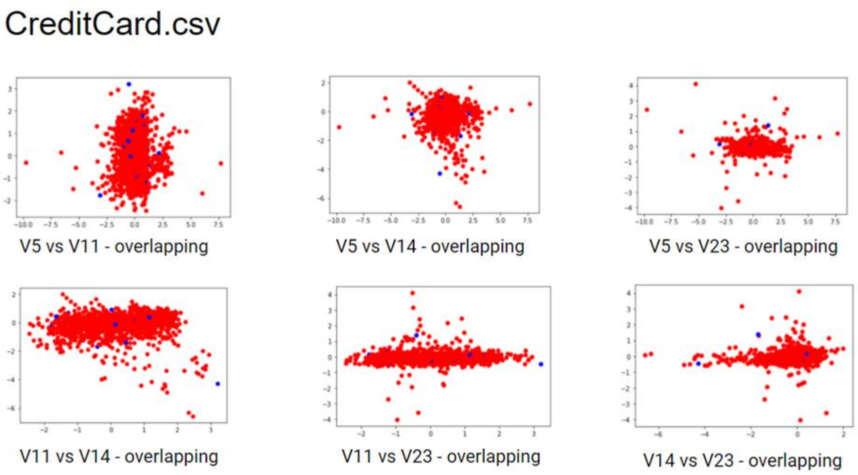
Understanding the Real-Life Datasets:

As far as the analysis with the real-life datasets is concerned, there are two such datasets – a ***CreditCard*** dataset and a ***Diabetes*** dataset. Both the datasets are imbalanced datasets, and can be found at [10] and [11] respectively.

The ***CreditCard*** dataset (saved as ***CreditCard.csv***) has 1550 instances, 30 feature variables (Time, *V1, V2, V3, …, V28, Amount*), and one target variable called 'Class', which corresponds to absence (Class = 0) of credit fraud, and the presence (Class = 1) of credit fraud. Class = 0 corresponds to 1525 instances, and Class = 1 corresponds to 25 instances, indicating the presence of <u>high imbalance</u> in the dataset.

The ***Diabetes*** dataset (saved as ***Diabetes.csv***) has 768 instances, 8 feature variables (*Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age*), and one target variable called 'Outcome', which corresponds to absence (Outcome = 0) of diabetes, and the presence (Outcome = 1) of diabetes. Class = 0 corresponds to 500 instances, and Class = 1 corresponds to 268 instances, indicating the presence of <u>moderate imbalance</u> in the dataset.

A graphical visualization of the levels of overlapping and separation, for all the 6 combinations of four random feature variables (*V5, V14, V11, V23* for the *CreditCard* dataset, and *Glucose, BloodPressure, BMI, DiabetesPedigreeFunction* for the *Diabetes* dataset) for the two given datasets is gives follows:



V5 vs V11 - overlapping

V5 vs V14 - overlapping

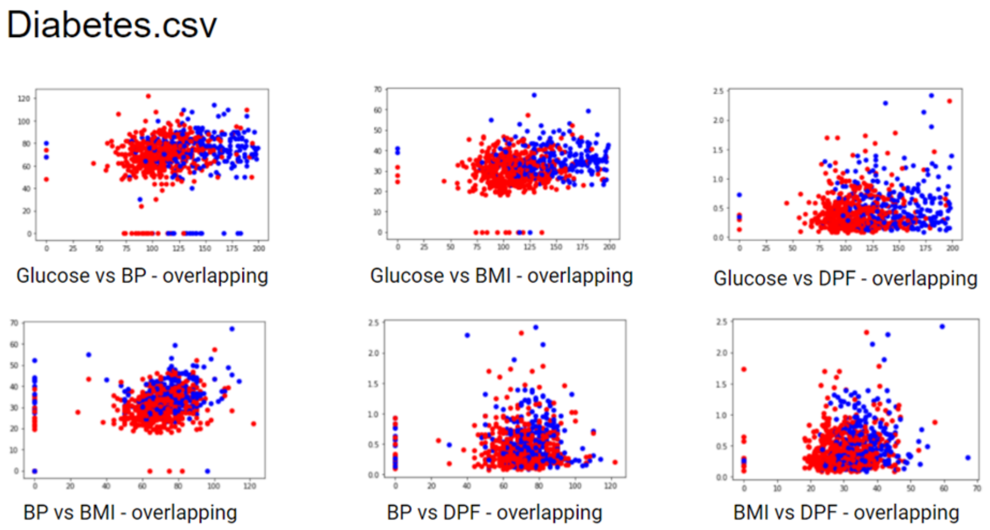
V5 vs V23 - overlapping

V11 vs V14 - overlapping

V11 vs V23 - overlapping

V14 vs V23 - overlapping

Fig.10 – The plot for one feature vs another, covering all 6 combinations of the four randomly selected feature variables (V5, V11, V14, V23) of the highly imbalanced and overlapping *CreditCard* Dataset. Here, all the four selected features heavily overlap, and the number of blue dots (corresponding to Class=1) is much less as compared to the red dots (corresponding to Class=0), which suggests the presence of <u>heavy imbalance</u> in the given dataset.



Diabetes.csv

Glucose vs BP - overlapping

Glucose vs BMI - overlapping

Glucose vs DPF - overlapping

BP vs BMI - overlapping

BP vs DPF - overlapping

BMI vs DPF - overlapping

Fig.11 - The plot for one feature vs another, covering all 6 combinations of the four randomly selected feature variables (*Glucose, BloodPressure, BMI, DiabetesPedigreeFunction*) of the moderately imbalanced and overlapping *Diabetes* Dataset. Here, all the four selected features heavily overlap, and the number of blue dots (corresponding to Class=1) is considerably less as compared to the red dots (corresponding to Class=0), which suggests the presence of moderate imbalance in the given dataset. In the given plots, *BP* corresponds to *BloodPressure*, and *DPF* corresponds to *DiabetesPedigreeFunction*.

We have used only four feature variables for showing the different combinations of the levels of imbalance and overlapping in the above plots, because the number of combinations increases as we increase the number of feature variables for the plots, thus providing us with a huge number of graphs, which would not be feasible for us to display within this research paper.

Since random feature variables are selected for both the datasets for plotting the above graphs, and all of them show high overlapping with significant imbalance, we can draw a universal conclusion, that for the datasets, all the feature variables are highly overlapping with respect to each other, and significant class imbalance exists in both the datasets.

Analysis of the Imbalanced and Overlapping CreditCard and Diabetes Datasets:

At first, we perform the Statistical Consistency Analysis for both the datasets, with respect to the two classes (0 and 1) present in the target variable, and understand the differences which are obtained. For ease of understanding, the results obtained by our model, for the *Diabetes dataset* (with a smaller number of feature variables), are tabularized as follows:

Table 9 - Statistical Consistency Analysis for the *Diabetes Dataset.*

Diabetes Dataset	Outcome	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes Pedigree Function	Age
N	0	500	500	500	500	500	500	500	500
	1	268	268	268	268	268	268	268	268
Mean	0	3.2980	109.9800	68.1840	19.6640	68.7920	30.3042	0.4297	31.1900
	1	4.8657	141.2575	70.8246	22.1642	100.3358	35.1425	0.5505	37.0672
Variance	0	9.085196	681.995600	325.622144	221.267104	9754.796736	59.015602	0.089273	135.861900
	1	13.944642	1016.332967	460.174468	311.405881	19162.902150	52.553862	0.138131	119.853698
Skewness	0	1.11075999	0.17259135	-1.80439066	0.03106169	2.49123855	-0.66390267	2.00021791	1.5668906
	1	5.00925330e-01	-4.92779129e-01	-1.93273775e+00	1.15259759e-01	1.83349545e+00	5.93487277e-04	1.71271794e+00	5.78385316e-01

The above Statistical Consistency Analysis of the *Diabetes* dataset, using some standard statistical measures like mean, variance, and skewness demonstrates that the presence of a significant skewness value in the overlapping features (i.e. all the features), renders it difficult to train the model due to the presence of the mixed boundary between the classes.

As we proceed further, the main motive of our study becomes the quantitative analysis of the two given real-life datasets using three performance indicators, namely 'Accuracy', 'Precision, and 'Recall'. We proceed with the same method, as used for the synthetic datasets i.e., finding the *Confusion Matrix,* from which we get the *TN, FP, FN,* and *TP,* and finally use these values to calculate the values of *Accuracy, Precision,* and *Recall.*

Training the Data for the Two given Real-Life Datasets, and Subjecting it to Different Algorithms for Analysis:

Each of the two real-life datasets are split into 75% training data and 25% testing data, and trained using 10-folds cross validation, before subjecting to the same machine learning algorithms, as for the synthetic datasets, namely *Decision Tree Classifier*, *K-Nearest Neighbour (KNN) Classifier*, and *Support Vector Machine (SVM) Classifier*, and two additional algorithms, namely *Logistic Regression,* and *Random Forest Classifier*. The performance indicator (*Accuracy*, *Precision*, *Recall*) values obtained for this model are then tabularised, and used for drawing necessary inferences.

Decision Tree Classifier:

The following table lists the performance indicator (*Accuracy*, *Precision*, and *Recall*) values obtained by our model, on applying the *Decision Tree Classifier*, on the two real-life datasets:

Table 10 - Tabularized data to show the performance indicators obtained for our model, when *Decision Tree Classifier* is applied to the *CreditCard* and *Diabetes* datasets.

Decision Tree Classifier	**Accuracy%**	**Precision%**	**Recall%**
CreditCard Dataset	96.21	0	0
Diabetes Dataset	71.01	56.46	56.76

K-Nearest Neighbour (KNN) Classifier:

The following table lists the performance indicator (*Accuracy*, *Precision*, and *Recall*) values obtained by our model, on applying the *KNN Classifier*, for K=3, on the two real-life datasets:

Table 11 - Tabularized data to show the performance indicators obtained for our model, when *KNN Classifier* is applied to the *CreditCard* and *Diabetes* datasets.

KNN Classifier	**Accuracy%**	**Precision%**	**Recall%**
CreditCard Dataset	98.37	0	0
Diabetes Dataset	69.08	56.78	51.76

Support Vector Machine (SVM) Classifier:

The following table lists the performance indicator (*Accuracy*, *Precision*, and *Recall*) values obtained by our model, on applying the *SVM Classifier*, on the two real-life datasets:

Table 12 - Tabularized data to show the performance indicators obtained for our model, when *SVM Classifier* is applied to the *CreditCard* and *Diabetes* datasets.

SVM Classifier	**Accuracy%**	**Precision%**	**Recall%**
CreditCard Dataset	98.37	0	0
Diabetes Dataset	75.35	73.00	46.24

Logistic Regression:

The following table lists the performance indicator (*Accuracy*, *Precision*, and *Recall*) values obtained by our model, on applying the *Logistic Regression*, on the two real-life datasets:

Table 13 - Tabularized data to show the performance indicators obtained for our model, when *Logistic Regression* is applied to the *CreditCard* and *Diabetes* datasets.

Logistic Regression	**Accuracy%**	**Precision%**	**Recall%**
CreditCard Dataset	98.19	0	0
Diabetes Dataset	76.04	70.24	55.24

Random Forest Classifier:

The following table lists the performance indicator (*Accuracy*, *Precision*, and *Recall*) values obtained by our model, on applying the *Random Forest Classifier*, on the two real-life datasets:

Table 14 - Tabularized data to show the performance indicators obtained for our model, when *Random Forest Classifier* is applied to the *CreditCard* and *Diabetes* datasets.

Random Forest	**Accuracy%**	**Precision%**	**Recall%**
CreditCard Dataset	98.37	0	0
Diabetes Dataset	74.28	68.03	53.71

_*Inference drawn from the above Classifiers (applied without resampling) for the real-life datasets:*

In the above implementations, five machine learning algorithms (*Decision Tree Classifier, KNN Classifier, SVM Classifier, Logistic Regression, Random Forest Classifier*) are applied on two real-life datasets, namely a *highly imbalanced CreditCard dataset*, and a *moderately Diabetes Dataset.*

For the *CreditCard dataset,* it is found that the model obtains accuracy rates of more than 96%, for all the five algorithms. This is suspicious, as it is difficult to get such high accuracy rates with large datasets. This behaviour can be attributed to the fact that the instances corresponding to *Class=0* are so many more in number as compared to the instances corresponding to *Class=1* i.e., presence of high imbalance, that in most of the cases the model predicts the value of an instance from the test set to have the Class as ‘0’, thus getting the overall accuracy score to be very high, even if it is predicting incorrectly for some instances. Aslo, for this dataset, we get the *precision (TP/(TP+FP))* and the *recall (TP/(TP+FN))* values to be zero, as we get the *TP* value as ‘0’ in the obtained *Confusion Matrix,* for all the five algorithms. This fact can also be used as a proof to support the reason behind such a high (although incorrect) accuracy rate, as obtained for the *CreditCard dataset*, as it shows that the model gives a very poor precision (~0). Thus, we can say that a correct classification would render lower *accuracy*, and higher *precision* and *recall* values for this dataset, which is exactly what is expected to happen when it is subjected to resampling techniques.

For the Diabetes dataset, all the five algorithms give very low performance indicator values (less than 80% for all algorithms, across all performance indicators). This can easily be attributed to the moderately imbalanced nature of this dataset, where the number of instances with *Outcome=1,* although considerably greater than those with *Outcome=0,* is not too high so as to lead the model to perform the classification of different instances into one class only. Although the model performs an honest classification, unlike that for the *CreditCard dataset*, the model makes mistakes at many points, which lowers the performance indicator values. Thus, a better performance would render higher performance indicator values, which is expected to happen when the dataset is subjected to resampling techniques.

Resampling the data for the CreditCard and Diabetes Datasets:

Since both the datasets are imbalanced datasets, resampling can be performed for improving the values of *Accuracy, Precision,* and *Recall.*

Just like we used SMOTETomek for resampling the synthetic datasets, we use it to resample the imbalanced *CreditCard* and *Diabetes* datasets.

The tables, for the performance indicator values, for the five algorithms, as obtained after applying SMOTETomek are as follows:

Decision Tree Classifier:

Table 15 - Tabularized data to show the performance indicators obtained for our model, when *Decision Tree Classifier* is applied to the *CreditCard* and *Diabetes* datasets after resampling them using *SMOTETomek*.

Decision Tree Classifier	**Accuracy%**	**Precision%**	**Recall%**
CreditCard Dataset	95.98	93.42	97.05
Diabetes Dataset	77.01	74.02	72.80

K-Nearest Neighbour (KNN) Classifier:

Table 16 - Tabularized data to show the performance indicators obtained for our model, when *KNN Classifier* is applied, for K=3, to the *CreditCard* and *Diabetes* datasets after resampling them using *SMOTETomek*.

KNN Classifier	**Accuracy%**	**Precision%**	**Recall%**
CreditCard Dataset	84.07	76.44	90.55
Diabetes Dataset	72.95	66.25	72.80

Support Vector Machine (SVM) Classifier:

Table 17 - Tabularized data to show the performance indicators obtained for our model, when *SVM Classifier* is applied to the *CreditCard* and *Diabetes* datasets after resampling them using *SMOTETomek*.

SVM Classifier	Accuracy%	Precision%	Recall%
CreditCard Dataset	64.58	57.93	61.92
Diabetes Dataset	76.13	77.64	61.60

Logistic Regression:

Table 18 - Tabularized data to show the performance indicators obtained for our model, when *Logistic Regression* is applied to the *CreditCard* and *Diabetes* datasets after resampling them using *SMOTETomek*.

Logistic Regression	Accuracy%	Precision%	Recall%
CreditCard Dataset	76.54	72.01	73.84
Diabetes Dataset	76.81	76.17	66.39

Random Forest Classifier:

Table 19 - Tabularized data to show the performance indicators obtained for our model, when *Random Forest Classifier* is applied to the *CreditCard* and *Diabetes* datasets after resampling them using *SMOTETomek*.

Random Forest Classifier	**Accuracy%**	**Precision%**	**Recall%**
CreditCard Dataset	99.01	98.43	99.27
Diabetes Dataset	82.55	80.68	76.00

Inference drawn from the above Classifiers (applied after resampling) for the real-life datasets:

After applying the same five classification algorithms on the two real-life datasets, after resampling them using SMOTETomek, we find that, for both the datasets, the performance indicator values turn out to be similar to what was expected.

For the *CreditCard datasets*, the accuracy values get reduced from the previously high values, and the *precision* and *recall* values get increased significantly, from the previous value of zero, as was expected from the effect of resampling. This change is very significant for the *KNN Classifier*, *SVM Classifier* and *Logistic Regression*. For the *Decision Tree Classifier*, the values have reduced slightly, and for the *Random Forest Classifier* the values have apparently increased. However, the overall readings suggest that the performance indicator values have reduced after resampling for the *CreditCard dataset*, as expected previously, thus indicating an improvement in correctness of the performance indicators, although the low values suggest that the model classifies the data more incorrectly than earlier.

For the Diabetes dataset, all the three performance indicator values have increased than earlier, thus suggesting better performance in classification after resampling. This is exactly what was expected for this dataset when resampling was to be performed.

CONCLUSION

As we can see in the results, the algorithms - *K Nearest Neighbour Classifier* and *Support Vector Machine Classifier* consistently provide similar and high-performance indicator values, with the *Decision Tree Classifier* lagging behind by a minimal percent, for both the synthetic and real-life datasets. On the real-life datasets, *Logistic Regression* gave moderate results, while Random *Forest Classifier* gave moderate-high results, irrespective of resampling.

The comparison of our results to the one in the paper proved ours to be better by a certain percent. Moreover, our implementation of SMOTETomek provided better results as compared to the previous ones.

The event where the worst accuracy and precision occurs for the synthetic datasets is when all the four features overlap for a balanced dataset followed by that for an unbalanced dataset. But the difference between the two happens to be quite large with that of the balanced dataset being on the lower end. This proves the statement that imbalance data has less overlapping than those of balanced. Also, for unbalanced data, using re-sampling may improve balance but does not guarantee better accuracy or precision.

For the real-life datasets, erroneously high results were obtained for the *CreditCard dataset* at first, which were later corrected to more realistic readings, after the application of resampling using SMOTETomek. On the other hand, for the *Diabetes dataset*, low performance indicator values were recorded in the beginning, which got improved later, thus suggesting improvement in performance, after the application of resampling.

Another noticeable result is that the use of SMOTETomek had the most impact on accuracy and precision for every algorithm in which there is a noticeable increase while there is minimal to none for the case of recall.

Although we believe that the implementation of these algorithms proved to be useful for the case of imbalance and overlapping, we believe that further in-depth prediction and analysis with different algorithms and techniques could lead to better results which may not be attained with the ones being used currently for both the synthetic and real-life datasets.

Since the problem of overlapping is still very underdeveloped and is not dealt with very much, there is still a lot of grounds which can be covered on this problem, which could lead to various inferences.

REFERENCES

1. T. Yaohua and G. Jinghuai. Improved classification for problem involving overlapping patterns. IEICE Transactions on Information and Systems, 90(11):1787–1795, 2007.
2. I. Tomek. Two modifications of CNN. IEEE Trans. Sys., Man and Cybernetics, 6:769–772, 1976
3. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16:321–357, 2002.
4. G. Batista, R. C. Prati, and M. C. Monard. Balancing strategies and class overlapping. In International Symposium on Intelligent Data Analysis, pages 24–35. Springer, 2005.
5. M. Denil and T. Trappenberg. Overlap versus imbalance. In Canadian Conference on Artificial Intelligence, pages 220–231. Springer, 2010.
6. G. Lemaˆıtre, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. The Journal of Machine Learning Research, 18(1):559–563, 2017.
7. H. Xiong, J. Wu, and L. Liu. Classification with class overlapping: a systematic study. In The 2010 International Conference on E-Business Intelligence, pages 491–497, 2010.
8. W. A. Almutairi. http: // www. cas. mcmaster. ca/ ~ cs3sd3/ waleed-data/ Data . 2019.
9. V. Lopez, A. Fern´andez, S. Garc´ıa, V. Palade, and F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences, 250:113–141, 201
10. Credit card dataset: https://www.kaggle.com/mlg-ulb/creditcardfraud
11. Diabetes dataset: https://www.kaggle.com/saurabh00007/diabetescsv