TGANs with Machine Learning Models in Automobile Insurance Fraud Detection and Comparative Study with Other Data Imbalance Techniques

Rohan Yashraj Gupta, Satya Sai Mudigonda, Pallav Kumar Baruah

Abstract: ***A data-driven Fraud detection model for insurance business can be seen as a two-phase method. Phase I is data-preprocessing of a given dataset, in which, handling class imbalance is a major challenge. Phase II is that of classification using Machine Learning models. It is important to comprehend if there is any influence of the technique used in Phase I on the efficiency of the model used for Phase II. A natural query that intrigues one is whether there is a golden combination of a technique in Phase I and a specific model in Phase II for assured best performance of a Fraud Detection Model.In this work, we study a few techniques for handling data imbalance issue namely, SMOTE, MWMOTE, ADASYN and TGAN in combination with various classifier models like Random Forest (RF), Decision Trees (DT), Support Vector Machines (SVM), LightGBM, XGBoost and Gradient Boosting Machines (GBM). The study is conducted on a dataset for motor vehicle insurance fraud detection.We present a comparison of various combinations of data imbalance technique and classifier models. It is observed that the combination of TGAN in Phase I and GBM in Phase II gives the best performance. This combination performs best in terms of important metrics such as false positive rate, precision and specificity. We obtained the lowest false positive rate of 0.0011 and precision of 0.9988 which minimizes the most critical risk for the insurance company of falsely classifying a non-fraud claim as a fraud. Finally, the specificity of 0.9989 indicates that the model was also very good at predicting the non-fraudulent claim.***

Keywords: Fraud Detection, Data Imbalance Techniques, Insurance Fraud, Machine Learning, Synthetic Data Generation, Class Imbalance.

I. INTRODUCTION

Fraud detection is an important problem and is taking priority in an insurance organization. According to the Federal Bureau of Investigation, the total fraud in non-health insurance is estimated to be around $40 Billion every year [1]. The Coalition Against Insurance Fraud (CAIF) report states that about $6 Billion is lost every year towards workers' compensation insurance fraud [2]. According to the study done by the Insurance Research Council (IRC), 15% to 17% of the total claims paid for automobile insurance (third-party) is fraudulent [3]. The study also estimated that around $7.7 Billion was added to auto insurance claims payment in the year 2012 compared to $5.8 Billion in 2002 [4]. These figures indicate that this is a major problem for any insurance business. They act as a major deterrent towards the growth. This is not a problem which is overlooked by any insurer. However, despite whatever measure they take to tackle fraud, fraudsters find a way. There are various antifraud measures that insurers' have. Most of them have SIUs (Special Investigation Units) which helps them to identify and investigate suspicious claims. According, to CAIF about 80% of the insurers had SUIs by 2001. They comprise of a small team who are trained to look for a routine type of fraud cases. This is a time taking process and a more complex fraud would go undetected in majority of the cases. However, in recent times with the advent of latest data science tools and techniques, insurers are finding ways to incorporate them into their existing processes. The traditional approach of detecting fraud using business rules are being augmented by data science tools and techniques. One major problem that is faced in building a fraud detection model for insurance is the data imbalance [5]. Lesser fraudulent data inhibits the model from getting trained better therefore the predictions are skewed towards the non-fraudulent claims. The model becomes very bad at predicting fraudulent claims, which is undesirable. In this work, we are have implemented tabular generative adversarial network based oversampling with machine learning models in automobile insurance fraud detection and performed a comparative study with various other machine learning models. The diagrammatic representation of the proposed model for fraud detection is shown in Figure 1.The work is divided into seven sections. Section 2 contains the literature survey. Section 3 explains the methodology adopted for this work. Section 4 covers the architecture of the TGANs model. Section 5 presents various performance metrics that are used to evaluate the model along with the business interpretation. Section 6 details the results obtained in the work. Section 7 contains the conclusion. Section 8 talks about future work. The work ends with appendix, acknowledgement and references.

Manuscript received on September 20, 2020.
Revised Manuscript received on January 24, 2021.
Manuscript published on January 30, 2021.
* Correspondence Author
Rohan Yashraj Gupta*, Department of Mathematics and Computer Science, Sri Sathya Sai Institute of Higher Learning, Puttaparthi, India. Email: rohanyashrajgupta@sssihl.edu.in
Satya Sai Mudigonda, Department of Mathematics and Computer Science, Sri Sathya Sai Institute of Higher Learning, Puttaparthi, India. Email: satyasaibabamudigonda@sssihl.edu.in
Pallav Kumar Baruah, Department of Mathematics and Computer Science, Sri Sathya Sai Institute of Higher Learning, Puttaparthi, India. Email: pkbaruah@sssihl.edu.in

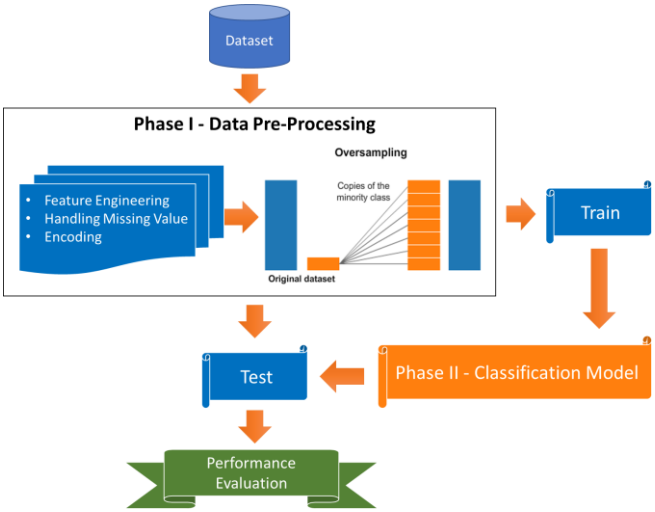

Figure 1 - Fraud detection model

II. LITERATURE SURVEY

This section contains various works that are present in the area of automobile insurance fraud detection. Various methods have been proposed by researchers.

Social Network Analysis (SNA) is one method on which various works can be found. Bodaghi and Teimourpour, n.d. [6] have proposed a new approach for fraud identification by making use of structural aspects of the social network. They use algorithms like DFS and BFS trees for cycle detection. Šubelj, Furlan, and Bajec have proposed a novel algorithm, Iterative Assessment Algorithm (IAA), which explores the relation between entities. Garcia and Hugo Proença, n.d. have proposed a variation of Subelj approach where they first randomly generate data that represents realistic data. After this, they apply the variation to the generated dataset. The major contribution is the approach to generate automobile data which can be used for social network analysis.

Other methods are used like random rough subspace-based neural network ensemble method [7]. Multistage methodology for insurance fraud detection committed by providers and patients as proposed by Johnson and Nagarur [8]. Zhao and Rousu have used clustering and generative methods for anomaly detection for patient visit data [9]. Unsupervised spectral ranking for anomaly detection in automobile insurance claims demonstrated by Nian et al. [10]. Verma, Taneja, and Arora have used rule-based pattern mining in insurance dataset for fraud detection [11].

Synthetic data is useful for training machine learning model as shown by Howe et.al. Generating such data tackles data imbalance which is one major problem faced during insurance fraud detection [12]. Sundarkumar, Ravi, and Siddeshwar have used a one-class support vector machine for undersampling the dataset [13]. Sundarkumar and Ravi improved this undersampling by employing k-Reserve Nearest Neighbourhood along with one-class support vector machine [14]. Subudhi and Panigrahi [15], [16]have demonstrated the use of ADASYN as class imbalance handling techniques along with SVM as classifiers to be good for an automobile insurance fraud detection. Itri et al. [17] have used random forest with k-measure evaluation for fraud detection in automobile insurance. Nikhil et. al. have used MWMOTE as an oversampling technique for automobile insurance claims [18]. Gupta RY et. al. have used SMOTE for oversampling the fraud cases [19], [20]. Recently, several GAN models have been developed to handle tabular data. RGAN and RCGAN [21], [22] can generate real-valued time-series data. medGAN [23], corrGAN [24] and several improved models [25]–[29] can generate discrete medical records but do not tackle the complexity in generating multimodal continuous variables. ehrGAN [30] generates augmented medical records but doesn't explicitly generate synthetic data. The tableGAN [31] tries to solve the problem of generating synthetic data for a tabular dataset which uses convolutional neural networks. TGANs [32] is a variation to this model which uses a recurrent neural network. In this work, we are using TGANs for data oversampling with machine learning models in automobile insurance fraud detection.

III. METHODOLOGY

The proposed methodology for fraud detection model has two major phases as shown in Figure 1.In phase I, data pre-processing is performed. In this feature engineering, handling missing values and data encoding is done. This phase is critical as it ensures that data is ready for the model. Once this is done the minority samples in the dataset is oversampled, this ensures that there is enough fraudulent data for the model to be trained on. The data is then split into train and test. In phase II, the training data is used for building a classification model. Once the model is built on the training data, the performance of the model is tested using the test dataset.

A. Data Description

The dataset used in this work is an automobile insurance dataset "carclaims.txt", which is publically available and is provided by Angoss Knowledge Seeker. The dataset has 32 features in a total of which 6 are ordinal, 25 are categorical and finally class variable which contains the label – "fraud" or "not-fraud". It consists of 15,420 records of which only 6% (923 records) are fraudulent, thereby making the dataset highly imbalanced in nature.

B. Data Pre-processing

1) Feature Engineering

Three features in the dataset are engineered using the available features. The first feature was accident date, which was derived using four feature – "Year", "Month" (month in which accident took place), "Week of month" (accident week of month) and "Day of week" (accident day of week). The second feature was claim reported date which was calculated using "Year", "Month Claimed" (Claim month), "Week of month claimed" (Claim week of month), "Day of week claimed" (Claim day of the week). Finally, using accident date and reported date a new feature called reporting delay was calculated as the difference between reported date and accident date. This allowed us to remove the dependent feature, thereby reducing the total number of features.

2) Handling Missing Value

After this, missing values were identified. There was only one record out of 15,420 records which had few of the features as missing, thus the record was deleted for the experimental purposes.

3) Data Encoding

Finally, categorical features were one-hot encoded [33], [34]. This was done to ensure that the data was ready for the following phase. However, before one-hot encoding, some of the features which had a very high number of categorical variables were reduced by clubbing together some of the variables into one. E.g. in the field "vehicle price", all the vehicles with the price 20,000 to 39,000 were put in one category.

4) Handling Data Imbalance

To handle the data imbalance, four most common methods were used – Synthetic Minority Oversampling Technique (SMOTE) [35], Majority Weighted Minority Oversampling Technique (MWMOTE) [36], Adaptive synthetic sampling approach for imbalanced learning (ADASYN) [15], [37] & Tabular Generative Adversarial Networks (TGAN) [32], [38], [39].

Using each of these methods, the fraud cases were generated such that the final ratio of fraud to non-fraud was 50:50 (approx.). This is an important step in this model, as this ensures that there is sufficient fraud data for the model to train with. In the later section, we will see the drastic improvement this phase bring in classification.

C. Classification model

After the data pre-processing phase, the data is divided into train and test in the ratio of 70:30. The training data is used for training the classifier. Finally, the efficacy of the classifier is tested using the test dataset and various performance metrics were compared.

In this work, we have used six classifiers – Decision Trees (DT), Random Forests (RF), Support Vector Machine (SVM), XGBoost, LightGBM and Gradient Boosting Machine (GMB).

We have compared these classifiers to find the best model for building a fraud detection model for automobile insurance.

In the following section, we briefly describe the TGAN architecture.

IV. ARCHITECTURE OF TGAN

This section contains the details of various components of TGAN architecture [32], [39]. Generating synthetic minority samples requires the TGAN model to be fitted to the existing data. The TGAN model used in this work requires the input data to have no missing values, a column of types integer, float variable, string or Boolean and finally each of the columns must contain data of only one type.

TGANs: The framework for Generative Adversarial Networks was first proposed by Ian J. Goodfellow [38]. GAN comprises of two models: a generative model G which generates synthetic data and a discriminative model D which discriminates between real and synthetic data. Both models are trained simultaneously. Statistically, G aims to maximize the probability of D making a mistake (refer Figure 2) [38].

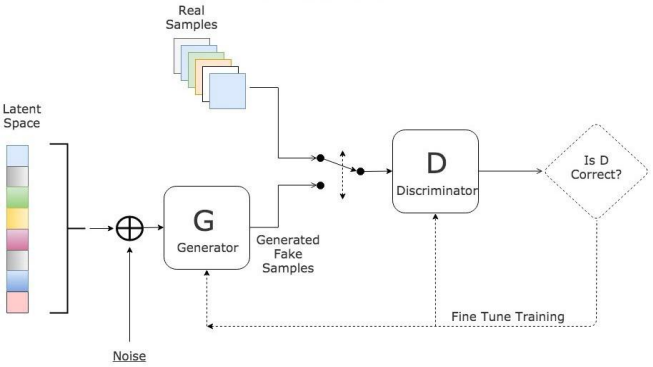

Figure 2 - GAN training network as depicted by Jonathan Hui

Xu and Veeramachaneni have developed a general-purpose GAN for generating tabular dataset, which they named it as Tabular GAN (TGAN) [32]. Figure 3 represents the TGAN architecture. The goal here is to develop a model G, such that the samples generated from this model when used to generate a machine learning model would produce similar accuracy as would the model learned using a real dataset.

Data transformations: Consider a table T consisting of n_c continuous variables and n_d categorical variables. Each claims record in the data is represented by a row in the table which is sampled independently. The data needs to undergo several transformations to input into the model.

Preprocessing of input variables: The numerical variables follows the multi-modal distribution. Gaussian Mixture Models (GMM) are used to sample values from multi-modal distribution. Thus, the transformations performed in cases of numerical variables becomes the output of the generator and input for the discriminator. [40]

Generator: They generate a numerical variable in 2 steps. First, generate the value scalar V, then generate the cluster vector U eventually applying tanh. To generate the desired row LSTM with attention mechanism is used. Input for LSTM in each step is random variable z, weighted context vector with previous hidden and embedding vector.

Discriminator: Multi-Layer Perceptron (MLP) with LeakyReLU [41] and Batch- Norm [42] is used. The first layer used concatenated vectors (V, U, D) among with mini-batch diversity with feature vector from LSTM. The loss function is the KL divergence term of input variables with the sum ordinal log loss function.

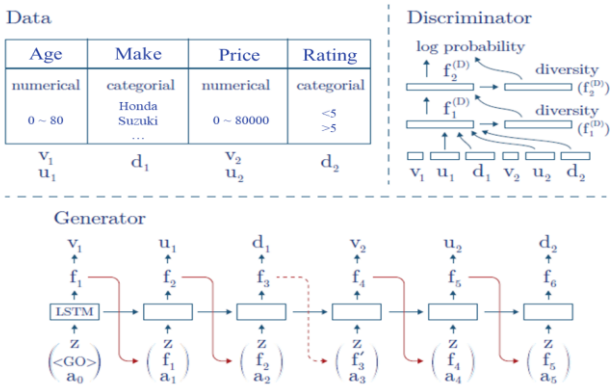

Figure 3 - TGAN Architecture

Model Parameters

The default behaviour of the TGAN Model can be changed by supplying different parameters such as batch size and the number of epochs [32], [39]. There are 12 different parameters for the model, of which the most important one which determines the performance of the model are 6 – Max epoch, steps per epoch, batch size, l2 norm, learning rate, and optimizer. The parameters used in this work is shown in Table 1. These parameters were selected based on the work of (L. Xu and Veeramachaneni 2018; Ashrapov 2020) [32], [39]

Table 1 - Model Parameters

Parameters	Values used
Max epoch	5
Steps per epoch	1,000
Batch size	200
Dimension in noise input for the generator	100
Noise	0.2
L2-Norm (regularization coefficient)	0.00001
Learning rate	0.001
No. of units in RNN cell	400
No. of units in fully connected layer	100
No. of layers in the discriminator	2
No. of units per layer in the discriminator	200
Optimizer	Adam Optimizer

V. PERFORMANCE METRICS

Performance metrics are used to measure the efficacy of the model. This section explains the various performance metrics that were used in this work to test the model. Business interpretation of the model is also given alongside [17], [43], [44].

Sensitivity = Fraud claims identified as fraud / Total fraud claims (actual)

This is the ratio of the fraud claims correctly identified as fraud to the total fraud claims in the dataset. An organization would want this number to be as close to value 1 as possible because they would want to identify as many fraudulent cases as possible. Also known as recall or hit rate.

Specificity = Non-fraud claims identified as non-fraud / Total non-fraud claims (actual)

This is the ratio of the non-fraud claims correctly identified as non-fraud to the total non-fraudulent claims in the dataset. An organization would want this number to be as close to value 1 as possible because the lower value would indicate that there is a higher number of non-fraud claims identified as fraud (false positive). This is unwanted because investigating a claim is an expensive procedure

Precision = Fraud claims identified as fraud / Total claims identified as fraud by the model

This measure the ratio of the true fraud claims to all the claims identified as fraud by the model. An organization would want this number to be as close to value 1 as possible because the lower value would indicate that there is a higher number of non-fraud claims identified as fraud (false positive). This is unwanted because investigating a claim is an expensive procedure.

Negative Predictive Value = Non-fraud claims identified as non-fraud / Total claims identified as non-fraud by the model

This measure the ratio of non-fraud claims correctly identified as non-fraud to all the claims identified as non-fraud by the model.

False Positive Rate = Non-fraud claims identified as fraud / Total non-fraud claims (actual)

This is the ratio of the non-fraud claims identified as fraud to total non-fraud claims. Insurers would want this value to be as close to value 0 as possible because a higher value would indicate that more number of non-fraud cases are identified as fraud, which is undesirable. Also known as fall-out.

False Discovery Rate =Non-fraud claims identified as fraud / Total claims identified as fraud by the model

This is the ratio of total non-fraud claims misclassified as fraud by the model to total claims identified as fraud by the model. This value should be closer to value 0 for a good fraud detection model.

False Negative Rate = Fraud claims identified as non-fraud / Total fraud claims (actual)

This is the ratio of fraud cases identified as non-fraud out of the total fraud cases. This indicated the percentage of the actual fraud cases which goes unidentified by the model. A good model would have this value as close to value 0 as possible. However, in reality, some of the fraud cases are intentionally left unidentified to prohibit further repercussion in the business process. Also known as miss-rate.

Accuracy = Total correct predictions both fraud and non-fraud / Total claims

This represents ratio the total correct prediction (both fraud and non-fraud) made by the model to the total claims in the dataset.

F1 Score = 2 × Precision × Recall / Precision + Recall

This is the harmonic mean of precision and sensitivity. This metric combines both precision and recall.

Values for a good fraud detection model is shown in the Table 2 [17], [43], [44]

Table 2 – Expected values for a good fraud detection model

Performance Metrics	Expected value
Sensitivity	> 0.95
Specificity	> 0.95
Precision	> 0.95
Negative Predictive Value	> 0.95
False Positive Rate	< 0.05
False Discovery Rate	< 0.05
False Negative Rate	< 0.05
Accuracy	> 0.95
F1 Score	> 0.95

VI. RESULTS AND DISCUSSION

This section details the results of the work carried out. As mentioned, in the earlier section, four techniques were used to handle data imbalance and six classifiers were used for classifying claims as fraud or non-fraud.

In the beginning, the data was preprocessed without handling the data imbalance. The data was then split into train and test (refer Table 3). The model was trained and tested on these datasets and the results were observed for the classifier model.

Table 3 - Unbalanced dataset

	Fraud	Non-Fraud	Total
Number	923	14,497	15,420
% of total	6%	94%	100%

The table containing performance metrics of various models are shown in Table 5. For the ease of comparison, each of the models is labelled from M1 to M30.

Figure 4 shows the plot of various metrics for all the models from M1 to M30. As mentioned earlier, False Positive Rate, False Discovery Rate and False Negative Rate are expected to have the values closer to 0 and the remaining metrics should have the values closer to 1 for an ideal fraud detection model.

From Figure 4 and Table 5 we can observe that M1 to M6 performs very poor compared to others. The precision of all these models is very low. XGBoost and GBM had the value greater than 0.8, this appears good but we observe that the sensitivity of these model are also low whereas specificity is very high. This means that majority of the predictions made by the model was the non-fraud case. The main reason for this behaviour of the model can be attributed to the fact that the model was trained on the data which had a majority of the data as non-fraud. This resulted in the predictions of the model to be skewed.

In the next step, the dataset was balanced using SMOTE, MWMOTE, ADASYN and TGAN. Each of the data imbalance techniques has a different way of generating synthetic data. The objective is to see, synthetic data generated from which of the four techniques would eventually help the classifier to train better thereby performing better. Table 4 shows the fraud and non-fraud claims after the balancing was done.

Table 4 - Balanced dataset

		Fraud	Non-Fraud	Total
SMOTE	**Number**	14,496	14,496	28,992
	% of total	50%	50%	100%
MWMOTE	**Number**	14,496	14,496	28,992
	% of total	50%	50%	100%
ADASYN	**Number**	14,528	14,496	29,024
	% of total	50%	50%	100%
TGAN	**Number**	14,323	14,496	28,819
	% of total	50%	50%	100%

The balanced data was now split into train and test and the model was trained and tested. Now, since there were more fraudulent claims for the model to be trained on, the prediction is better, which is reflected in the results obtained. However, among all these, there were few models which perform comparatively better than others.

From the results, few models stand out compared to others in terms of various performance metrics (refer Table 2) – M8, M11, M14, M23, M26, M28, M29 and M30.

It can be observed that the poorest performing classifier is SVM (M9, M15, M21 and M27) among all the remaining classifiers. Models with poor performance were M16, M17 and M18. M16, M17 and M18 were XGBoost, LightGBM and GBM respectively trained on MWMOTE generated dataset. Here it is important to note that M14, which is a random forest classifier trained on MWMOTE generated dataset, is among the best performing model in terms of sensitivity, accuracy and F1-Score. However, when we look at classifiers trained on TGAN generated data, we observe an improved performance of the model in comparison to others. When compared to all the models M26, M28, M29 and M30 were among the best performing model. These were Random Forest, XGBoost, LightGBM and GBM classifiers trained on TGAN generated dataset.

Moreover, M30 is the best in terms of Specificity, Precision, False Discovery Rate, and False Negative Rate with comparable accuracy and F1-score. The model gave the highest precision of 0.9988. This minimizes the risk for the insurance company of falsely classifying a non-fraud claim as fraudulent. Also, the lowest false positive rate indicates that very less non-fraudulent claims were classified as fraudulent by the model.

From the results, it is observed that on this dataset, some models perform best in some metrics while others perform best in others. If we have a different dataset, say health insurance, which has a completely different property, the conclusion may differ. Hence we cannot say one model is the best for fraud detection. However, from an automobile insurance companies' perspective, M30 would be the best model. As this model would minimize the risk for an insurance company of falsely classifying a non-fraud claim as fraudulent compared to other models. It is also observed that SVMs are bad in this classification exercise, this can be attributed to the fact that the data is not linearly separable [45].

The appendix contains the ROC curves for models M7 to M30.

Table 5 - Results of the fraud detection model

Data Imbalance Technique	Classifier	Model No	AUC-ROC	Sensitivity	Specificity	Precision	Negative Predictive Value	False Positive Rate	False Discovery Rate	False Negative Rate	Accuracy	F1 Score
-	**DT**	M1	0.5613	0.1829	0.9396	0.1711	0.9440	0.0604	0.8289	0.8171	0.8913	0.1768
	RF	M2	0.5012	0.0081	0.9942	0.0870	0.9363	0.0058	0.9130	0.9919	0.9313	0.0149
	SVM	M3	0.5000	0.0000	1.0000	0.0000	0.9362	0.0000	0.0000	1.0000	0.9362	0.0000
	XGBoost	M4	0.5100	0.0203	0.9997	0.8333	0.9374	0.0003	0.1667	0.9797	0.9372	0.0397
	LightGBM	M5	0.5112	0.0244	0.9981	0.4615	0.9375	0.0019	0.5385	0.9756	0.9359	0.0463
	GBM	M6	0.5121	0.0244	0.9997	0.8571	0.9376	0.0003	0.1429	0.9756	0.9375	0.0474
SMOTE	**DT**	M7	0.9335	0.9388	0.9281	0.9297	0.9375	0.0719	0.0703	0.0612	0.9335	0.9342
	RF	M8	0.9616	0.9542	0.9689	0.9688	0.9543	0.0311	0.0312	0.0458	0.9615	0.9614
	SVM	M9	0.7668	0.9506	0.5830	0.6977	0.9211	0.4170	0.3023	0.0494	0.7679	0.8047
	XGBoost	M10	0.8990	0.9493	0.8487	0.8640	0.9429	0.1513	0.1360	0.0507	0.8993	0.9046
	LightGBM	M11	0.9643	0.9394	0.9892	0.9887	0.9416	0.0108	0.0113	0.0606	0.9641	0.9634
	GBM	M12	0.9033	0.9432	0.8634	0.8748	0.9376	0.1366	0.1252	0.0568	0.9036	0.9077
MWMOTE	**DT**	M13	0.9642	0.9997	0.9287	0.9341	**0.9997**	0.0713	0.0659	**0.0003**	0.9644	0.9658
	RF	M14	**0.9900**	**0.9989**	0.9811	0.9817	0.9989	0.0189	0.0183	0.0011	**0.9901**	**0.9902**
	SVM	M15	0.7772	0.9268	0.6277	0.7159	0.8944	0.3723	0.2841	0.0732	0.7781	0.8078
	XGBoost	M16	0.7962	0.9421	0.6502	0.7316	0.9174	0.3498	0.2684	0.0579	0.7970	0.8236
	LightGBM	M17	0.8751	0.9761	0.7740	0.8139	0.9697	0.2260	0.1861	0.0239	0.8757	0.8876
	GBM	M18	0.7969	0.9402	0.6535	0.7331	0.9152	0.3465	0.2669	0.0598	0.7977	0.8238
ADASYN	**DT**	M19	0.9328	0.9418	0.9238	0.9264	0.9397	0.0762	0.0736	0.0582	0.9329	0.9341
	RF	M20	0.9581	0.9544	0.9619	0.9623	0.9539	0.0381	0.0377	0.0456	0.9581	0.9583
	SVM	M21	0.7658	0.9462	0.5854	0.6993	0.9144	0.4146	0.3007	0.0538	0.7675	0.8042
	XGBoost	M22	0.9021	0.9514	0.8528	0.8682	0.9451	0.1472	0.1318	0.0486	0.9026	0.9079
	LightGBM	M23	0.9633	0.9358	0.9908	0.9905	0.9381	0.0092	0.0095	0.0642	0.9631	0.9624
	GBM	M24	0.9092	0.9481	0.8703	0.8817	0.9427	0.1297	0.1183	0.0519	0.9096	0.9137
TGAN	**DT**	M25	0.9480	0.9503	0.9458	0.9436	0.9522	0.0542	0.0564	0.0497	0.9480	0.9469
	RF	M26	0.9664	0.9403	0.9924	0.9916	0.9457	0.0076	0.0084	0.0597	0.9670	0.9653
	SVM	M27	0.7997	0.9269	0.6724	0.7297	0.9061	0.3276	0.2703	0.0731	0.7967	0.8166
	XGBoost	M28	0.9705	0.9423	0.9986	0.9985	0.9477	0.0014	0.0015	0.0577	0.9711	0.9696
	LightGBM	M29	0.9701	0.9423	0.9978	0.9976	0.9477	0.0022	0.0024	0.0577	0.9707	0.9692
	GBM	M30	0.9705	0.9420	**0.9989**	**0.9988**	0.9475	**0.0011**	**0.0012**	0.0580	0.9711	0.9696

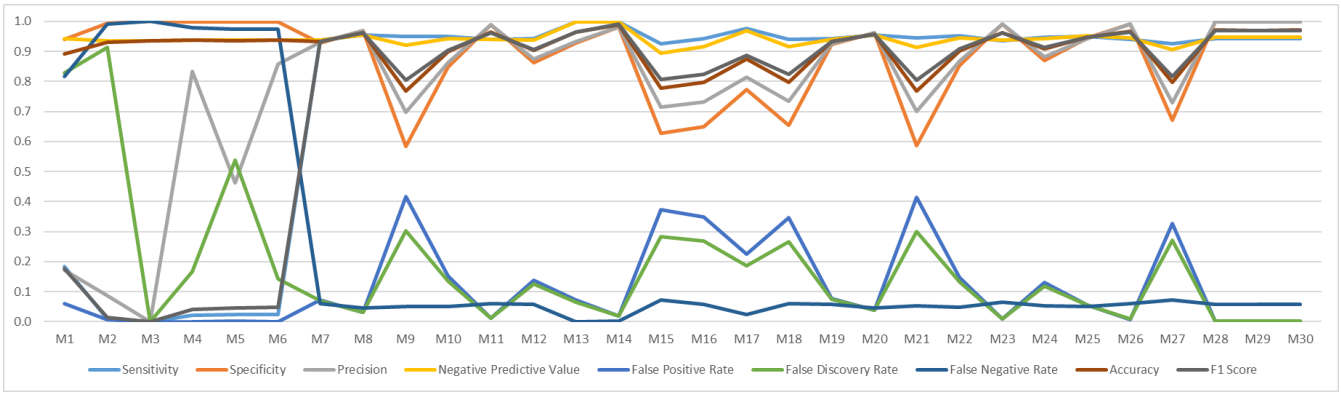

Figure 4 - Plot of performance metrics for all the models

VII. CONCLUSION

In this work, we have performed a comparative study of various fraud detection model built using the most commonly used classifiers. Detecting fraud in insurance is extremely challenging. We have used various data imbalance handling techniques in combination with various machine learning-based classification model. The overall objective of this work is to arrive at the most effective fraud detection tools for automobile insurance. The work has been evaluated on a globally available dataset, "carclaims.txt". The results show that the best performing model was TGAN+GBM with the best specificity, precision and false discovery rate. Also, it is observed that the TGAN generated data was much better for model training compared to the rest of the class imbalance handling techniques.

FUTURE WORK

The work can be extended to other lines of business like health insurance, to find the best-suited combination of data imbalance technique and classification model. Further, data embedding can be performed after phase I of the method proposed in this work.

APPENDIX

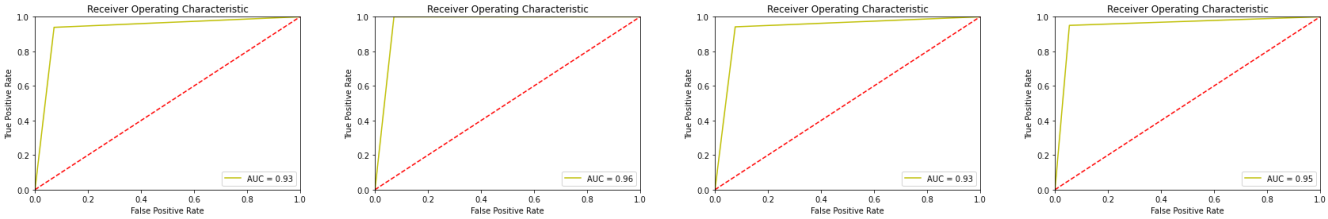

Figure 5 - ROC curves of DT with SMOTE, MWMOTE, ADASYN & TGAN

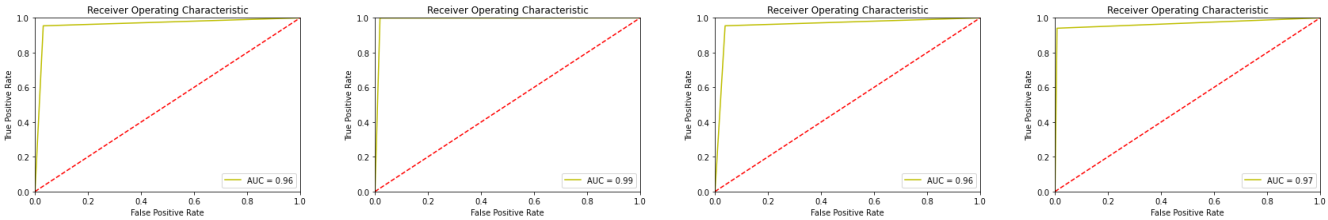

Figure 6 - ROC curves of RF with SMOTE, MWMOTE, ADASYN & TGAN

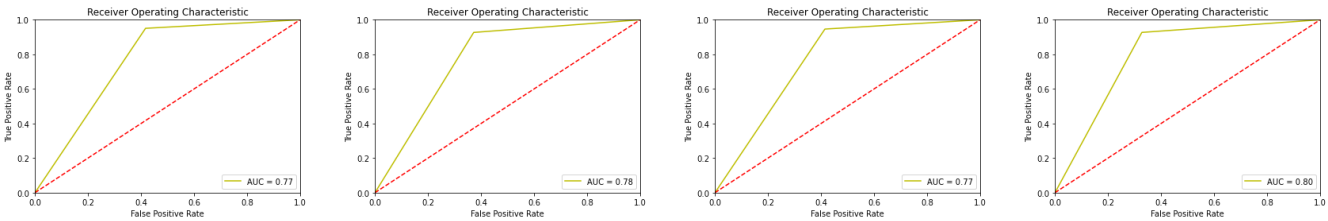

Figure 7 - ROC curves of SVM with SMOTE, MWMOTE, ADASYN & TGAN

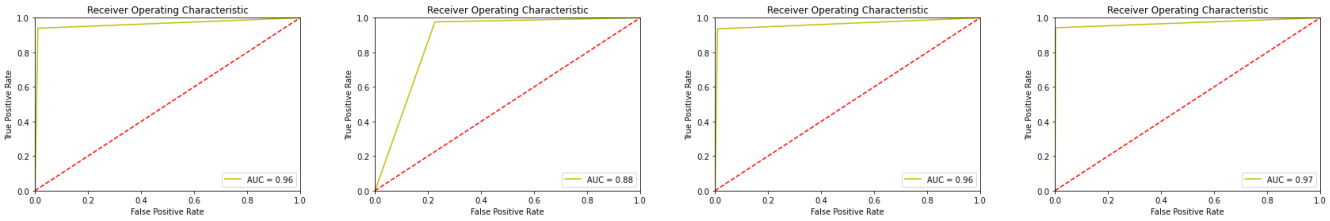

Figure 8 - ROC curves of LightGBM with SMOTE, MWMOTE, ADASYN & TGAN

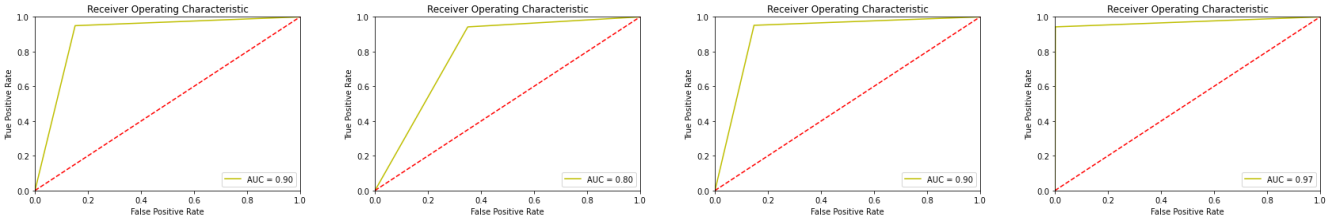

Figure 9 - ROC curves of XGBoost with SMOTE, MWMOTE, ADASYN & TGAN

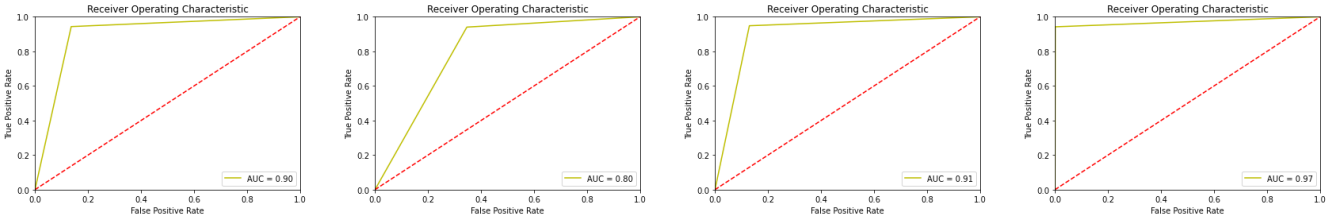

Figure 10 - ROC curves of GBM with SMOTE, MWMOTE, ADASYN & TGAN

ACKNOWLEDGMENT

We dedicate this work to the Revered Founder Chancellor of Sri Sathya Sai Institute of Higher Learning, Bhagawan Sri Sathya Sai Baba.

REFERENCES

1. "Insurance Fraud — FBI." [Online]. Available: https://www.fbi.gov/stats-services/publications/insurance-fraud. [Accessed: 29-Jan-2020].
2. "Coalition Against Insurance Fraud: Rapid National Response Urged To Head Off Coming Wave of COVID-19 Insurance Scams - InsuranceNewsNet." [Online]. Available: https://insurancenewsnet.com/oarticle/coalition-against-insurance-fraud-rapid-national-response-urged-to-head-off-coming-wave-of-covid-19-insurance-scams#.Xts8DzozbIX. [Accessed: 06-Jun-2020].
3. "Research Publications | Insurance Research Council." [Online]. Available: https://www.insurance-research.org/research-publications/6. [Accessed: 15-Jan-2021].
4. Corum, "Insurance Research Council Finds That Fraud and Buildup Add Up to $7.7 Billion in Excess Payments for Auto Injury Claims," *Insur. Res. Counc.*, p. 3, 2015.
5. N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study1," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Nov. 2002, doi: 10.3233/IDA-2002-6504.
6. A. Bodaghi and B. Teimourpour, "The detection of professional fraud in automobile insurance using social network analysis."
7. W. Xu, S. Wang, D. Zhang, and B. Yang, "Random Rough Subspace Based Neural Network Ensemble for Insurance Fraud Detection," in *2011 Fourth International Joint Conference on Computational Sciences and Optimization*, 2011, pp. 1276–1280, doi: 10.1109/CSO.2011.213.
8. M. E. Johnson and N. Nagarur, "Multi-stage methodology to detect health insurance claim fraud," *Health Care Manag. Sci.*, vol. 19, no. 3, pp. 249–260, Sep. 2016, doi: 10.1007/s10729-015-9317-3.
9. Y. Zhao and J. Rousu, "Anomaly Detection from Patient Visit Data," 2016.
10. K. Nian, H. Zhang, A. Tayal, T. Coleman, and Y. Li, "Auto insurance fraud detection using unsupervised spectral ranking for anomaly," *J. Financ. Data Sci.*, vol. 2, no. 1, pp. 58–75, Mar. 2016, doi: 10.1016/j.jfds.2016.03.001.
11. A. Verma, A. Taneja, and A. Arora, "Fraud detection and frequent pattern matching in insurance claims using data mining techniques," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, 2017, vol. 2018-Janua, no. August, pp. 1–7, doi: 10.1109/IC3.2017.8284299.
12. A. K. I. Hassan and A. Abraham, "Modeling Insurance Fraud Detection Using Imbalanced Data Classification," in *Advances in Intelligent Systems and Computing*, vol. 419, Springer Verlag, 2016, pp. 117–127.
13. G. G. Sundarkumar, V. Ravi, and V. Siddeshwar, "One-class support vector machine based undersampling: Application to churn prediction and insurance fraud detection," in *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2015, no. ii, pp. 1–7, doi: 10.1109/ICCIC.2015.7435726.
14. G. G. Sundarkumar and V. Ravi, "A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance," *Eng. Appl. Artif. Intell.*, vol. 37, pp. 368–377, Jan. 2015, doi: 10.1016/j.engappai.2014.09.019.
15. S. Subudhi and S. Panigrahi, "Effect of Class Imbalanceness in Detecting Automobile Insurance Fraud," in *2018 2nd International Conference on Data Science and Business Analytics (ICDSBA)*, 2018, pp. 528–531, doi: 10.1109/ICDSBA.2018.00104.
16. S. Subudhi and S. Panigrahi, "Use of optimized Fuzzy C-Means clustering and supervised classifiers for automobile insurance fraud detection," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 32, no. 5, pp. 568–575, Jun. 2020, doi: 10.1016/j.jksuci.2017.09.010.
17. B. Itri, Y. Mohamed, Q. Mohammed, and B. Omar, "Performance comparative study of machine learning algorithms for automobile insurance fraud detection," in *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, 2019, pp. 1–4, doi: 10.1109/ICDS47004.2019.8942277.
18. N. Rai, P. K. Baruah, S. S. Mudigonda, and P. K. Kandala, "Fraud Detection Supervised Machine Learning Models for an Automobile Insurance," *Int. J. Sci. Eng. Res.*, vol. 9, no. 11, pp. 473–479, 2018.

19. R. Y. Gupta, S. Sai Mudigonda, P. K. Kandala, and P. K. Baruah, "Implementation of a Predictive Model for Fraud Detection in Motor Insurance using Gradient Boosting Method and Validation with Actuarial Models," in *2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES)*, 2019, pp. 1–6, doi: 10.1109/INCCES47820.2019.9167733.
20. R. Y. Gupta, S. S. Mudigonda, P. K. Kandala, and P. K. Baruah, "A Framework for Comprehensive Fraud Management using Actuarial Techniques," *Int. J. Sci. Eng. Res.*, vol. 10, no. 3, pp. 780–791, 2019.
21. A. Sarraf and Y. Nie, "RGAN: Rényi Generative Adversarial Network," *SN Comput. Sci.*, vol. 2, no. 1, p. 17, Feb. 2021, doi: 10.1007/s42979-020-00403-9.
22. R. B. Arantes, G. Vogiatzis, and D. R. Faria, "rcGAN: Learning a Generative Model for Arbitrary Size Image Generation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, vol. 12509 LNCS, pp. 80–94, doi: 10.1007/978-3-030-64556-4_7.
23. K. Armanious *et al.*, "MedGAN: Medical image translation using GANs," *Comput. Med. Imaging Graph.*, 2020, doi: 10.1016/j.compmedimag.2019.101684.
24. G. Marti, "CORRGAN: Sampling Realistic Financial Correlation Matrices Using Generative Adversarial Networks," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2020, doi: 10.1109/ICASSP40776.2020.9053276.
25. L. Aviñó, M. Ruffini, and R. Gavaldà, "Generating synthetic but plausible healthcare record datasets," *arXiv*. 2018.
26. E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," *arXiv*. 2017.
27. B. K. Beaulieu-Jones *et al.*, "Privacy-preserving generative deep neural networks support clinical data sharing," *Circ. Cardiovasc. Qual. Outcomes*, 2019, doi: 10.1161/CIRCOUTCOMES.118.005122.
28. J. Walonoski *et al.*, "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record," *J. Am. Med. Informatics Assoc.*, 2018, doi: 10.1093/jamia/ocx079.
29. A. Yahi, R. Vanguri, N. Elhadad, and N. P. Tatonetti, "Generative adversarial networks for electronic health records: A framework for exploring and evaluating methods for predicting drug-induced laboratory test trajectories," *arXiv*. 2017.
30. Z. Che, Y. Cheng, S. Zhai, Z. Sun, and Y. Liu, "Boosting deep learning risk prediction with generative adversarial networks for electronic health records," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2017, doi: 10.1109/ICDM.2017.93.
31. N. Park *et al.*, "Data Synthesis based on Generative Adversarial Networks," *arXiv*, vol. 11, no. 10, pp. 1071–1083, Jun. 2018, doi: 10.14778/3231751.3231757.
32. L. Xu and K. Veeramachaneni, "Synthesizing Tabular Data using Generative Adversarial Networks," *arXiv*, Nov. 2018.
33. P. Cerda and G. Varoquaux, "Encoding high-cardinality string categorical variables," *arXiv*. 2019, doi: 10.1109/tkde.2020.2992529.
34. M. Guerzhoy, "One-Hot Encoding," *University of Toronto*. 2016.
35. N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.
36. S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE--Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014, doi: 10.1109/TKDE.2012.232.
37. Haibo He, Yang Bai, E. A. Garcia, and Shutao Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328, doi: 10.1109/IJCNN.2008.4633969.
38. I. J. Goodfellow *et al.*, "Generative Adversarial Networks," Jun. 2014.
39. I. Ashrapov, *Tabular GANs for uneven distribution*. 2020.
40. S. S. Mullick, S. Datta, and S. Das, *Generative Adversarial Minority Oversampling*, vol. 2019-Octob. Institute of Electrical and Electronics Engineers Inc., 2019, pp. 1695–1704.
41. B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," May 2015.
42. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Feb. 2015.
43. J. Lever, M. Krzywinski, and N. Altman, "Classification evaluation," *Nat. Methods*, 2016, doi: 10.1038/nmeth.3945.
44. N. Seliya, T. M. Khoshgoftaar, and J. Van Hulse, "A study on the relationships of classifier performance metrics," in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2009, doi: 10.1109/ICTAI.2009.25.
45. Z. Liu, X. Lv, K. Liu, and S. Shi, "Study on SVM compared with the other text classification methods," in *2nd International Workshop on Education Technology and Computer Science, ETCS 2010*, 2010, doi: 10.1109/ETCS.2010.248.

AUTHORS PROFILE

Rohan Yashraj Gupta, is a doctoral research scholar in Sri Sathya Sai Institute of Higher Learning in the field of Actuarial sciences. His area of research includes Data-Driven Fraud Detection and Prevention using Actuarial Techniques and Technology. He is an Actuarial data science expert and has published research papers in international journals.

Satya Sai Mudigonda, A Senior Tech Actuarial Consultant providing services in USA and India. With a wide skill set, he managed numerous multi-million-dollar international assignments for major insurance companies across the globe. He is an honorary professor in Sri Sathya Sai Institute of Higher Learning. He has published about fifteen papers in the field of Actuarial data science and has presented in several international conferences.

Dr. Pallav Kumar Baruah, is an Associate Professor and the former HOD of the Department of Mathematics and Computer Science of Sri Sathya Sai Institute of Higher Learning. He has guided several research scholars in mathematics and computer science. He has numerous research publications to his credit and has presented in several national and international conferences.