IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 November 2022, accepted 7 December 2022, date of publication 12 December 2022, date of current version 16 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3228507

RESEARCH ARTICLE

Evaluation of Synthetic Data Generation Techniques in the Domain of Anonymous Traffic Classification

DRAKE CULLEN[1], JAMES HALLADAY[1], NATHAN BRINER[1], RAM BASNET[1], JEREMY BERGEN[1], AND TENZIN DOLECK[2]

[1]Department of Computer Science and Engineering, Colorado Mesa University (CMU), Grand Junction, CO 81501, USA
[2]Faculty of Education, Simon Fraser University, Burnaby, BC V5A 1S6, Canada

Corresponding author: Ram Basnet (rbasnet@coloradomesa.edu)

This work was supported by the State of Colorado through funds appropriated for Cybersecurity Law Dubbed "Cyber Coding Cryptology for State Records."

ABSTRACT Anonymous network traffic is more pervasive than ever due to the accessibility of services such as virtual private networks (VPN) and The Onion Router (Tor). To address the need to identify and classify this traffic, machine and deep learning solutions have become the standard. However, high-performing classifiers often scale poorly when applied to real-world traffic classification due to the heavily skewed nature of network traffic data. Prior research has found synthetic data generation to be effective at alleviating concerns surrounding class imbalance, though a limited number of these techniques have been applied to the domain of anonymous network traffic detection. This work compares the ability of a Conditional Tabular Generative Adversarial Network (CTGAN), Copula Generative Adversarial Network (CopulaGAN), Variational Autoencoder (VAE), and Synthetic Minority Over-sampling Technique (SMOTE) to create viable synthetic anonymous network traffic samples. Moreover, we evaluate the performance of several shallow boosting and bagging classifiers as well as deep learning models on the synthetic data. Ultimately, we amalgamate the data generated by the GANs, VAE, and SMOTE into a comprehensive dataset dubbed CMU-SynTraffic-2022 for future research on this topic. Our findings show that SMOTE consistently outperformed the other upsampling techniques, improving classifiers' F1-scores over the control by ∼7.5% for application type characterization. Among the tested classifiers, Light Gradient Boosting Machine achieved the highest F1-score of 90.3% on eight application types.

INDEX TERMS Anonymous traffic, synthetic data, CopulaGAN, CTGAN, SMOTE, VAE, TabNet, deep learning, machine learning, unbalanced data.

I. INTRODUCTION

Network traffic often contains sensitive user data and private information, so the classification of this traffic is considered a controversial topic. Although network traffic classification can be used for censorship, it is also necessary for Internet Service Providers (ISPs) to guide initiatives such as resource allocation, infrastructure development, improving network security, and other network services [1]. Concerns regarding personal and professional security have led to the proliferation of many traffic anonymization technologies such as Tor, VPNs, Hypertext Transfer Protocol Secure (HTTPS), Transport Layer Security (TLS), and Secure Shell Protocol (SSH). Unfortunately, while these techniques increase privacy for the users, they also make it more difficult for ISPs to scale their network.

Tor and VPNs are some of the most common anonymization protocols. Tor traffic is encrypted through a series of network nodes that the client selects. Each node is encrypted with a unique key and can only interact with the previous and

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei.

next node. As the layers of encrypted nodes increase, tracing each node and the overall path back to its origin/destination is incredibly difficult, resulting in increased user privacy [2]. Similarly, VPNs establish encrypted connections from users' devices to remote servers. All internet traffic is routed through the secure connection with the external server. ISPs and other entities can no longer see the websites and services a user is connecting to, instead, all traffic is a connection to the VPN provider [3]. VPN traffic is dependent on the integrity of a third party service while Tor is a decentralized system that relies on a community of volunteers.

Classifying anonymized traffic is complicated by the fact that network traffic is inherently imbalanced [4]. While a webpage can be loaded with relatively few packets, Peer-to-Peer (P2P) and streaming services can exchange millions of packets. This means that models trained on network data may perform well during production, but their performance can fail to scale after deployment. Since poorly performing minority classes may be of primary interest to the ISP's future development, it is important to ensure that these models scale well to real-world applications [5].

Data generated through specifically designed algorithms, referred to as synthetic data, has been shown as a potential solution to the aforementioned imbalanced data problem. Synthetic Minority Oversampling Technique (SMOTE), one of the most prominent synthetic data generation techniques, was introduced in 2002 [6]. Due to the explosion in the application of deep learning, several generative models have also recently been introduced such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) in 2014 [7], [8]. Since these methods may capture different information from the original data to produce their synthetic samples, it makes sense that particular classification techniques may benefit more from different generative techniques.

In this article, we compare the efficacy of several data generation techniques and analyze their impact on performance across a spectrum of cutting-edge deep and shallow learning models for anonymous traffic classification. Moreover, we show that traffic classification models trained on imbalanced data fail to maintain performance when exposed to balanced traffic data, establishing poor performance for minority classes. Then we show this problem can be addressed when our originally imbalanced data is augmented with synthetic samples. We found deep learning models tended to experience greater variance in performance metrics from the GAN and VAE data, while both deep and shallow learning methods benefit from the data generated by SMOTE.

The existing need for balanced network data for anonymous traffic classification served as motivation for this work [5], [9], and [10]. By exploring various generative models, we hope to provide clarity on which synthetic data techniques could be applied in network traffic classification and demonstrate an effective methodology for doing so. The following points outline the novel contributions of this work:

- Assessment and comparison of four prominent synthetic data techniques (CTGAN, CopulaGAN, VAE, and SMOTE) and their ability to generate effective synthetic network traffic samples
- Application of generative techniques scarcely used in this domain, specifically a VAE and two state-of-the-art GAN variants
- Performance evaluation of many boosting, bagging, and deep learning models trained on synthetic network data
- Generation of a new synthetic anonymous network traffic dataset to balance the generally unbalanced network traffic data and enable future synthetic data research
- Improve performance for multiclass classification of eight application types

The remainder of this paper is presented in the following order: Section 2 explores works related to anonymous network traffic and synthetic data applications. Section 3 analyzes the CIC-Darknet2020 dataset [9] used in our experiments. Section 4 gives a brief description of the unique frameworks and architecture used for experimentation. Section 5 provides explainability to our CMU-SynTraffic-2022 dataset. Section 6 presents our experimental methodology, while section 7 discusses experimental results. Section 8 investigates the limitations of the present work and provides avenues for future research. Finally, section 9 concludes and summarizes the paper.

II. RELATED WORKS

A. VPN TRAFFIC DETECTION

Draper-Gil et al. [12] studied the ability for time-related features to detect encrypted communications utilizing VPN services. They created the ISCX-VPN2016 dataset consisting of traffic from several applications such as browsing and streaming traffic to conduct their experiments. Their classifiers, C4.5 and KNN, were trained to distinguish VPN and non-VPN traffic as well as to classify the traffic type. The paper found that the C4.5 classifier was slightly more effective, obtaining precision and recall scores around 84% when paired with the correct flow-timeout value.

Caicedo-Muñoz et al. [13] integrated a quality of service (QoS) classifier and per-hop behavior (PHB) to the ISCX-VPN2016 dataset. They generated two new datasets: the first dataset contained VPN and non-VPN traffic while the second dataset combined VPN and non-VPN traffic with PHB labels. Bagging and boosting algorithms were most effective on their datasets, achieving accuracies of 94.42% and 92.82%.

Miller et al. [14] captured real VPN and non-VPN network data using Wireshark and NetMate. TCP flow-based features were used to train a multi-layer perceptron classifier to detect OpenVPN and non-VPN traffic. Their neural network model achieved an accuracy of around 94% on the post-training test set.

B. TOR TRAFFIC DETECTION

Lashkari et al. [15] devised a two layer approach for detecting and classifying Tor traffic. They created a labeled Tor dataset named ISCX-Tor2016 that was published alongside their research. Furthermore, the team extracted time-based features from the dataset and used them as the sole features to train their models. Models were trained on data with different flow lengths, and they found 15 seconds to be the optimal flow time. Their top model achieved a recall and precision of 99% when detecting Tor traffic, and 83% when distinguishing between eight application types.

Huo et al. [16] noted that a large number of parameters need to be calculated to train a network to classify Tor traffic. As calculating these parameters is computationally expensive, they propose a new model that extracts spatial features by CNN layers, gathers temporal features from LSTM layers, then fuses multi-scale features before sending the features to an attention mechanism. They were able to achieve 94.9% accuracy on the ISCX-Tor2016 dataset.

Gurunarayanan et al. [17] performed random oversampling and random undersampling on the ISCX-Tor2016 dataset to detect Tor traffic. The team incorporated Grid Search algorithms as a means of hyperparameter tuning. Their top model–Random Forest–achieved an accuracy of 99%.

C. DATA AUGMENTATION

Jadav et al. [10] trained 15 machine learning classifiers on the CIC-Darknet2020 dataset to differentiate between darknet and benign traffic. They addressed a class imbalance problem between the number of darknet and benign samples by incorporating SMOTE and found that Extra Tree and Decision tree to be the highest performing classifiers with accuracies of 99%. They recommend future work to investigate deep learning models and multiclass classification with many possible classes.

Guo et al. [5] introduced Imbalanced Traffic Classification General Adversarial Network (ITCGAN) as a solution to imbalanced data in the domain of network traffic classification. Using the ISCX-VPN2016 dataset, they tested ITCGAN against other synthetic data generation techniques. Namely Random OverSampling, SMOTE, ADAptive SYNthetic algorithm, SMOTE+Support Vector Machine, SMOTE+Tomek Links, and a Conditional generative adversarial network all on a 1D-CNN classifier. ITCGAN, SMOTE, and SMOTE+SVM were the only oversampling methods that outperformed the baseline dataset.

Wang et al. [18] proposed a GAN based methodology named FlowGAN to address the problem of class imbalance in the field of network traffic classification. FlowGAN was trained on traffic from the ISCX-VPN2016 dataset. Real data and synthetic data generated from FlowGAN were concatenated and used to train a multilayer perceptron neural network. In comparison to the unbalanced dataset, FlowGAN increased F1-score by 15.6%. When compared to a balanced dataset, the F1-score increased by 2.12% on average.

Okonkwo et al. [19] applied Convolutional Neural Networks to encrypted traffic classification on ISCX-Tor2016 and ISCX-VPN2016 datasets. The data traffic flows are gathered and reconstructed into flow images, or flowpics. To balance the dataset, data augmentation is used on the flowpics to create new samples. Each flowpic was then sent to the CNN and then classified as either HTTPS, VPN, or Tor. Several more CNNs were trained for the tasks of classifying application identification and origin containing four and eight classes respectively. They obtained an average accuracy of ~93% across all experiments.

Li et al. [20] presented a data augmentation technique utilizing a GAN, VAE, and statistical parameter configuration (SPC) to address the problem of insufficient network data. The GANs data deviated from the actual traffic with a mean and variance both less than 1.7%. The proposed GAN technique outperformed both the SPC and VAE.

D. ANONYMOUS TRAFFIC DETECTION

Lashkari et al. [11] aggregated the CIC-Darknet2020 dataset by merging two encrypted traffic datasets and introduced DeepImage. DeepImage is a model that creates a gray image by selecting the most important features from a dataset. DeepImage sends the gray image to a two-dimensional convolutional neural network (CNN) to detect and categorize eight types of darknet traffic. Overall, the CNN had an accuracy of 86%; however, the CNN struggled to classify certain traffic types such as browsing traffic.

Gupta et al. [21] expanded upon the body of knowledge by training classifiers to detect three types of traffic: non-VPN/non-Tor traffic, Tor traffic, and VPN traffic. Previous research classified traffic as either VPN vs non-VPN, or Tor vs non-Tor, but not both. Eight machine learning algorithms were trained on the dataset, and XGBoost was the highest performer with an accuracy of 98%.

Iliadis et al. [22] trained five machine learning classifiers on the CIC-Darknet2020 dataset to detect and classify darknet traffic into one of four categories – Tor, non-Tor, VPN, and non-VPN. Their feature importance analysis found "Total Length of Fwd Packet" to be the most vital feature while they removed the five socket-related features to avoid overfitting. Random Forest performed the best with an accuracy of over 98% on darknet detection and classification.

Al-Omari et al. [23] analyzed the impact of training machine learning algorithms on unique feature groups to differentiate darknet and regular traffic. They settled on four feature groups: "all features" (68 features), "all features without Src Port and Dst Port" (66 features), "selected features" (9 features), "selected features without Src Port and Dst Port" (11 features). Overall, boosting algorithms tended to work the best, and the Ridge-300 classifier had an accuracy of 99.9% on the "selected features" feature set.

E. LIMITATIONS AND KNOWLEDGE GAP OF CURRENT WORKS

Table 1 outlines the current methodologies used to detect and classify network traffic. We find the following gaps in the existing research:

- Network traffic is frequently imbalanced and is an ongoing problem in network research [9]. While previous

TABLE 1. Summary of related works in network traffic classification.

Paper	Dataset	Classifier(s)	Multiclass	Synthetic Data
Draper-Gil et al. [12]	ISCX-VPN2016	C4.5, KNN	Yes (7)	No
Guo et al. [5]	ISCX-VPN2016	One-dimensional CNN	Yes (6)	Four SMOTE Variants, ROS, ITCGAN
Wang et al. [18]	ISCX-VPN2016	MLP	Yes (15)	FlowGAN
Lashkari et al. [15]	ISCX-Tor2016	Zero R, C4.5, KNN	Yes (8)	No
Li et al. [20]	Network Traffic	NA	6 Traffic Types	GAN, VAE, SPC
Gupta et al. [21]	CIC-Darknet2020	XGB, KNN, NB, RF, ADA, Balanced Bagging, ANN, LSTM	Yes (3)	No
Iliadis et al. [22]	CIC-Darknet2020	Multi-layer perceptron, DT, RF, KNN, GB	Yes (4)	No
Lashkari et al. [11]	CIC-Darknet2020	Two-dimensional CNN	Yes (8)	No
This Paper	CIC-Darknet2020	XGB, LGBM, RF, DNN, TabNet	Yes (8)	SMOTE, VAE, CopulaGAN, CTGAN

research [5], [18], [20] has applied synthetic data generation techniques to balance VPN, Tor, and internet traffic datasets separately; no research has compared the efficacy of many synthetic data generation models against each other on a robust anonymous network traffic dataset.

- Variational autoencoders have proven to be effective in generating synthetic network traffic [20]. No reviewed papers have used a VAE to generate anonymous synthetic traffic samples.
- No known research has applied the state of the art TabNet model [24] to classify anonymous traffic.
- Prior research does not evaluate the effect of synthetic anonymous data samples on the classification of audio streaming, browsing, chat, email, file transfer, p2p, video streaming, and VoIP traffic.

In the present paper, we bridge the aforementioned knowledge gaps by using VAE, GAN, and SMOTE algorithms to generate synthetic anonymous traffic for data balancing. These three data generation techniques have yet to be compared in this domain, so providing a direct comparison can indicate the preferred technique in anonymous network traffic classification scenarios. Moreover, we compare boosting and bagging classifiers to deep learning models in their classification effectiveness when trained on synthetic data.

III. DATASET

A. FEATURE EXTRACTION AND COMPOSITION

The CIC-Darknet2020 dataset provided by the Canadian Institute for Cybersecurity [11] is incorporated in our experiments. The data set was formed through the fusion of two public datasets, namely ISCX-Tor2016 [15] and ISCX-VPN2016 [12], to create an anonymous dataset encompassing regular, Tor, and VPN traffic. Furthermore, the dataset was published in both raw Packet-Capture (PCAP) files and tabular data that was preprocessed by CIC-FlowMeter v4.0 over a predetermined time interval. These tabular samples contain time-based features that capture statistics from the traffic flow such as flow duration and packet inter-arrival times. This is combined with information about the packets' source and destination, the flags declared in their headers, and the time at which the flow was captured, creating a well-encapsulated representation of the traffic flow. We chose this dataset because it contains a relatively large number of application types (8) and samples (117,620) while incorporating both Tor and VPN traffic.

A two-layered approach was used to generate data for the CIC-Darknet2020. Regular and anonymous traffic were synthesized in the first layer, and the traffic was further broken into eight application types: audio streaming (Vimeo and Youtube), browsing (Firefox and Chrome), chat (ICQ, AIM, Skype, Facebook and Hangouts), email (SMTPS, POP3S and IMAPS), file transfer (Skype, FTP over SSH (SFTP) and FTP over SSL (FTPS) using Filezilla and an external service), p2p (uTorrent and Transmission), video streaming (Vimeo and Youtube), and VoIP (Facebook, Skype and Hangouts voice calls) in the second layer. Figure 1 presents the traffic and application type sample ratios.

B. PREPROCESSING AND FEATURE ENGINEERING

Before generating synthetic data, we applied feature selection and data cleaning. Initially, samples containing Inf and NaN values were eliminated from the dataset. Of the 84 features in CIC-Darknet2020, 14 of them contained the value zero (0) for every sample in this dataset. These features were discarded as they do not contribute to the model performance in discriminating various traffic types. Six additional features (Flow-id, Source/Destination IP, Timestamp, and Source/Destination port) were eliminated from the dataset, bringing the total number of features down to 64. The Flow-id feature is of the form (Source IP)-(Destination IP)-(Source Port)-(Destination Port)-(Protocol), therefore it only contains duplicate information. Source/Destination IP is an artifact of the original dataset and does not represent the distribution of IP addresses found on the internet. The Timestamp feature was removed because our classifiers do not account for the order of flows and information about when a flow was initiated will not add any meaningful information. Similarly, Source/Destination

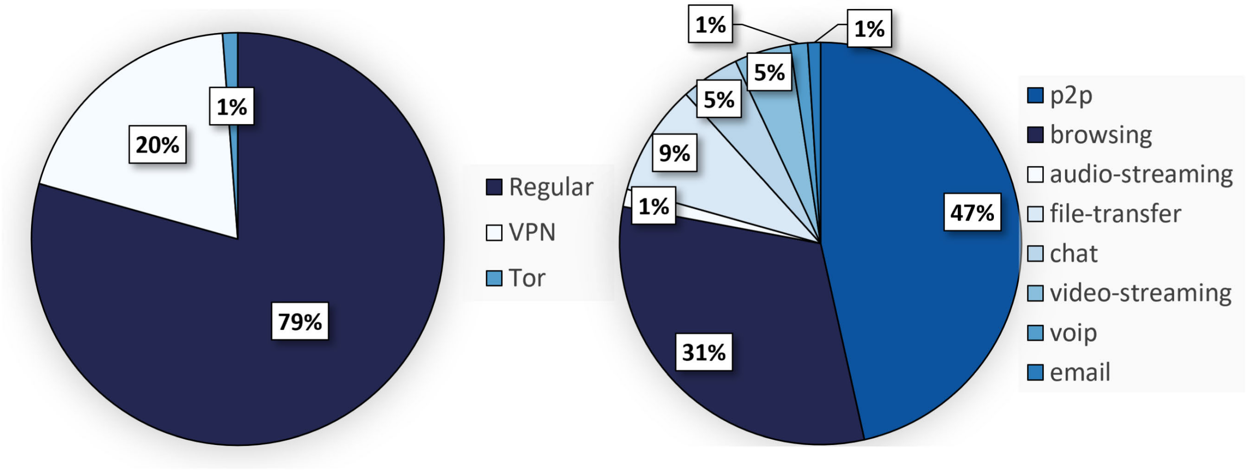

FIGURE 1. CIC-Darknet2020 traffic and application type sample ratios.

Ports are unique non-deterministic identifiers which could result in overfitting.

IV. FRAMEWORKS AND ARCHITECTURE

Our experiments incorporate three shallow learning and two deep learning classifiers. Sun et al. [25] note that shallow learning classifiers tend to be more effective at classifying structured data with XGBoost (XGB), Light Gradient Boosting Machines (LGBM), and Random Forest (RF) being top performers. On the other hand, TabNet is a cutting-edge deep learning model that warrants experimentation in the field of anonymous traffic detection. The following section will present the shallow learning classifiers, three synthetic data generation techniques, and an overview of TabNet's architecture.

A. RANDOM FOREST

RF is an ensemble shallow learning classifier composed of a series of decision trees. Each decision tree in the RF model adapts the divide-and-conquer paradigm. Data splits occur at each internal node and decisions are reached at the leaf nodes. RF incorporates techniques such as bagging and randomness to aggregate the predictions of the decision trees and reduce variance and bias that may occur in a single decision tree [26].

B. LIGHTGBM

Microsoft introduced LGBM–a gradient-boosted decision tree–in 2016 as a high-speed tree-based model that can handle large data by growing trees vertically. LGBMs predecessors tended to be much slower because they use information gain as a guiding heuristic to conduct optimal splits through the use of pre-sorted or histogram-based algorithms. LGBM addresses this concern by using Exclusive Feature Bundling (EFB) and Gradient One-Side Sampling (GOSS). EFB combines mutually exclusive features and GOSS randomly drops small gradients because larger gradients are usually associated with greater information [27].

C. XGBOOST

XGB is a gradient-boosted decision tree algorithm built on top of the Gradient Boosting Machine's (GBM) framework. XGB optimizes the GBM framework by filling in missing data through the process of sparsity awareness, addressing the overfitting problem by lowering variance and increasing bias through Lasso Regression, using the weighted quantile sketch algorithm to find optimal tree splits, performing depth-first tree pruning, parallelized decision tree construction, and out-of-core computing [28].

D. SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE (SMOTE)

SMOTE [6] is an oversampling approach that generates samples for minority classes. SMOTE begins by selecting a random sample from the minority class and its k-nearest neighbors (neighbors that reside in the same feature space). Out of the k neighbors, a random neighbor is selected and the distance between the two points is calculated. The distance is multiplied by a random value between 0 and 1 and added to the feature vector to generate a new sample. The process is repeated until a satisfactory number of samples have been generated. Many techniques have been explored such as the adaptive synthetic (ADASYN) sampling approach as potential improvements to the original SMOTE algorithm [29].

E. GENERATIVE ADVERSARIAL NETWORK (GAN)

$$\begin{aligned} min_G max_D V(D, G) &= \mathbb{E}_{x \sim p_{data}(x)}[log(D(x))] \\ &\quad + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z))) \\ D &= Discriminator \quad G = Generator \\ x &= input \quad z = noise\ input \\ p_{data}(x) &= data\ distribution \\ p_z(z) &= noise\ distribution \end{aligned} \tag{1}$$

A GAN model is composed of two independent models: a generator and a discriminator. The generator and discriminator make up an adversarial network where the generator attempts to synthesize new samples and the discriminator works to identify the synthetic samples. After training, the generator will be able to create samples from noise input that will preserve and correspond to the distribution of the training data [8]. This process was mathematically modeled in [6, eq. (1)]. This equation is a value function in which the generator, G, is trying to minimize the function and the discriminator, D, is trying to maximize the function. $E_{x \sim p_{data}}$ and $E_{z \sim p_z(z)}$ are the expected value for an input of original data and an input of noise respectively. The rest of this equation is adapted from the binary cross entropy function used to model binary classification problems.

Conditional Tabular GAN (CTGAN) is a GAN-inspired architecture that is capable of generating tabular data. CTGAN improves existing models such as table-GAN [30] by incorporating the variational Gaussian mixture model for each column rather than normalizing continuous values between -1 and 1 [31], [32]. CopulaGAN [33] is a variation of CTGAN where the Cumulative Distribution Function transformation is applied via GaussianCopula and it attempts to learn column correlation in a table [34], [35].

F. VARIATIONAL AUTOENCODER (VAE)

$$\begin{aligned} \mathcal{L}(\theta, \phi; x_i) &= \mathbb{E}_{q_\phi(z|x_i)}[log\, p_\theta(x_i|z)] \\ &\quad - D_{KL}(q_\phi(z|x_i)||p_\theta(z)) \\ z &= latent\ representation \quad x = input \\ q &= encoder \quad p = decoder \\ \theta &= encoder\ weights \\ \phi &= decoder\ weights \end{aligned} \tag{2}$$

A VAE [7] is an autoencoder that specializes in reducing overfitting through regularization. Standard autoencoders work by encoding data into a smaller feature space (with minimal information loss) and then utilizing a decoder to reconstruct an output that is as similar as possible to the original data. The output from the decoder is compared to the initial data and weights are updated through backpropagation to minimize future reconstruction errors. Since autoencoders attempt to train an encoder and decoder with as little loss as possible, they are susceptible to overfitting. Variational Autoencoders address this concern by encoding the input as a distribution over the latent space [36]. After a VAE is trained it can be used to generate new synthetic samples for a dataset.

(2) is the loss function for a variational autoencoder and consists of two terms [5]. The first term, $E_{z \sim q(z|xi)}[log\, p_\phi(x_i|z)]$, models the reconstruction loss i.e. a measure of how similar a reconstructed output sample is to the input. The second term is the Kullback-Leibler (KL) divergence which evaluates the difference between two distributions. Minimizing the KL regularizes the output probability distribution of the encoder. More explicitly, reducing kl divergence ensures the latent distribution matches a normal distribution.

$$let\ z \sim p(z|x) = \mathcal{N}(\mu, \sigma^2), \tag{3}$$

$$then\ z = \mu + \sigma * \epsilon \tag{4}$$

$$where\ \epsilon \sim \mathcal{N}(0, 1)$$

Optimizing this loss function can prove problematic as finding the gradient of this equation is not possible in its current form. This is because we are sampling from a random node which results in an intractable integral. To address this problem, Kingma et al. [5] describe the reparameterization trick. The reparameterization trick is a tool to backpropagate when sampling a random node from a distribution (in the loss function, looking at $q_\phi(z|x_i)$, z is a random variable sampled from a distribution and is the problematic term). To represent z in a deterministic way, it can be written as $z = \mu + \sigma * \epsilon$, where ϵ is a predetermined sample from a separate distribution $p(\epsilon) = N(0, 1)$. By substituting z with this new representation, it becomes possible to evaluate the gradient of the loss function.

G. TABNET

Arik et al. [24] recognized that deep learning models are effective in fields such as image recognition. Since a large portion of existing data is arranged in a tabular format, they proposed the deep tabular data learning architecture named TabNet in late 2020. Through the process of sequential attention, TabNet acts similarly to decision trees while adding interpretability and more efficient learning. At each step, features are passed through a feature transformer composed of a fully connected layer, batch normalization, and a Gated Linear Unit. Next, the features are sent to an attentive transformer made up of a fully connected layer, batch normalization, and sparse max normalization. The attentive transformer considers feature importance from previous steps to create a mask. The mask determines which features are most suitable to be used by the model. The mask improves model interpretability because it shows which features TabNet deemed to be the most important [24].

V. CMU-SYNTRAFFIC-2022

To facilitate future research on synthetic data and anonymous network traffic, our team has produced the CMU-SynTraffic-2022 dataset containing synthetic data generated in our experiments as well as the real data used to generate it. The synthetic portion of this dataset consists of 432,847 SMOTE, 700,000 CTGAN, 700,000 CopulaGAN and 700,000 VAE samples. CMU-SynTraffic-2022 also contains 117,620 real samples from CIC-Darknet2020 [11] for a total of 2,650,467 samples. In addition to the 64 features present in CIC-Darknet2020, this dataset also contains the data source label (real, CTGAN, CopulaGAN, VAE, SMOTE). The traffic, application, and data source is further visualized in Figure 2.

Our team performed four tests as a preliminary measure of the synthetic data using the SDV Framework [37]: Logistic

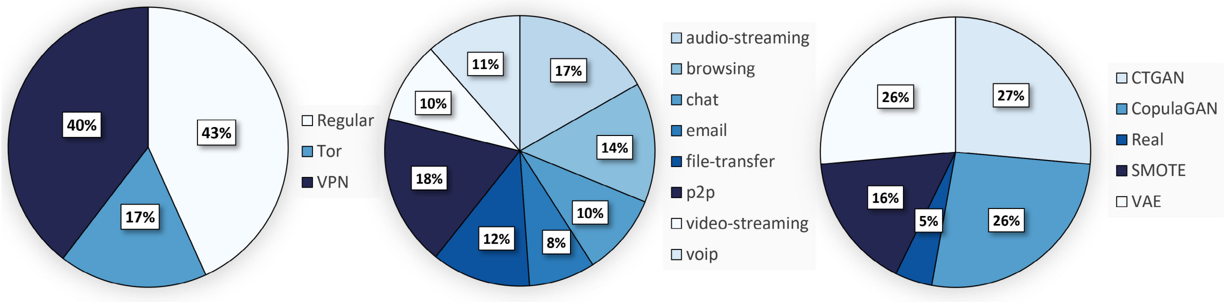

FIGURE 2. Sample ratios for traffic type, application type, and data source for CMU-SynTraffic-2022.

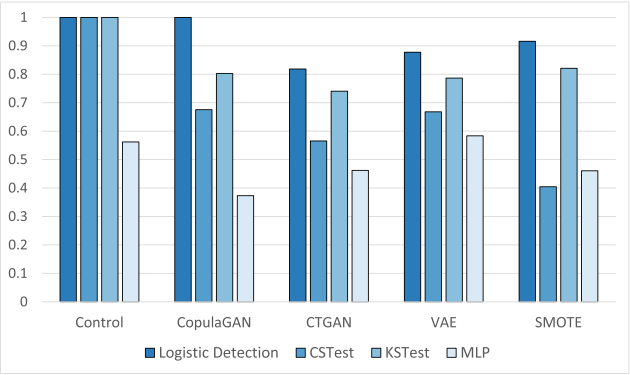

FIGURE 3. CMU-SynTraffic-2022 synthetic evaluation metrics.

Detection, chi-squared test (CS Test), Kolmogorov-Smirnov test (KS Test), and Multilayer perceptron (MLP) classifier test. The CS and KS tests determine how closely correlated the distributions of the synthetic data is to the real data [38], [39]. Logistic Detection trains a classifier to differentiate real and synthetic data and MLP is simply the F1-Score of a multilayer perceptron trained on the synthetic data when classifying by application type [40].

One noteworthy observation as seen in Figure 3 is that the CopulaGAN model achieved higher logistic detection, CS test, and KS test metrics compared to the other models. This indicates that the distribution of the CopulaGAN synthetic data may more closely match real data. Conversely, CopulaGAN had the lowest MLP F1-score among the models with the VAE model performing the best, even outperforming the original dataset.

VI. EXPERIMENT METHODOLOGY

A. OVERVIEW

Our experiments were conducted in correspondence with Figure 4. Detailed discussions about data collection, data cleaning, and feature selection are presented in the Dataset section. Subsequent sections will outline the experiment scenarios and the remaining stages of the research methodology.

B. EXPERIMENT SCENARIOS

Our experiments are conducted in two distinct scenarios. Each scenario begins by establishing baseline results by training 2 boosting classifiers, 1 bagging algorithm, and two deep neural networks on an imbalanced dataset consisting of real data. These classifiers are tasked to classify samples from two separate test sets. One of the test sets contains imbalanced sample proportions, while the other test dataset is balanced. The contrast between classifier performance on the imbalanced and the balanced test sets is meant to showcase that classifiers trained on heavily imbalanced datasets may perform poorly when deployed in real-world applications where traffic may not be skewed in the same manner as the training data. These results will be dubbed Imbalanced Control and Balanced Control respectively. Next, an upsampled dataset composed of synthetic and real data is utilized to train the same five classifiers. Performance metrics were gathered to determine whether classifiers trained on synthetic data can differentiate various anonymous traffic types and to determine whether training on synthetic data increases classifier performance.

Scenario A emphasizes high-level classification among Tor, VPN, and regular traffic where each traffic type is composed of a basket of eight application types. Scenario B aims to differentiate among eight application types. Both Scenario A and B are trained on the original imbalanced dataset as well as the upsampled datasets using synthetic data generation techniques.

C. TEST AND SEED SPLIT

Before generating synthetic data, the original dataset was divided into seed and test datasets for Scenario A and Scenario B. The Scenario A test dataset consists of 1,950 samples (650 of each traffic type) and the Scenario B test dataset is composed of 4,000 samples (500 samples of each application type). The remaining samples for both scenarios are present in their respective seed datasets. The datasets were limited by the fact that there were only 742 Tor samples (Scenario A) and 572 email samples (Scenario B), so we chose these proportions to ensure that the test sets are balanced while leaving enough samples in the seed dataset for reliable synthetic data generation.

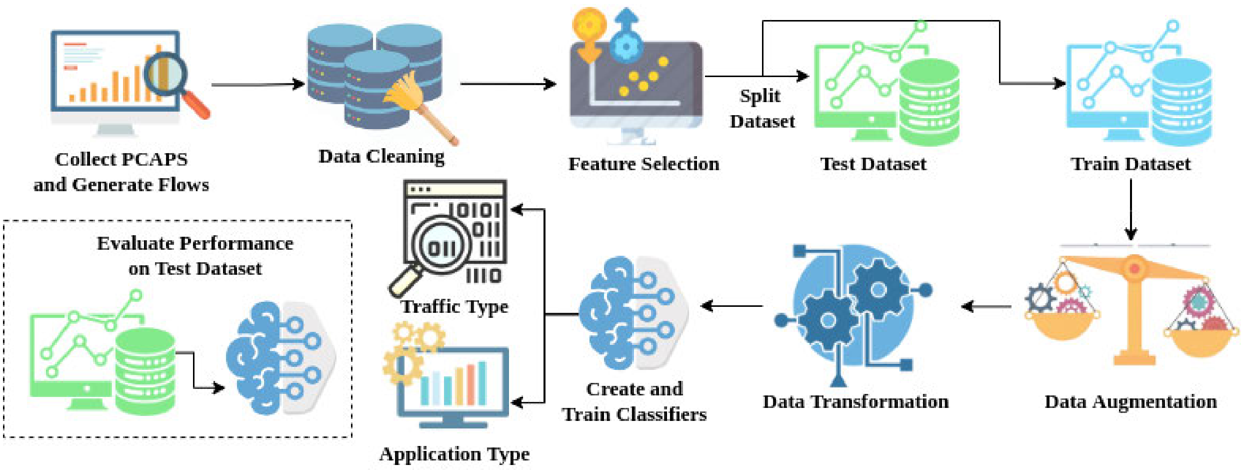

FIGURE 4. Experimental workflow.

The justification for splitting data into a test and seed dataset prior to generating synthetic data is twofold. First, by splitting the dataset in two, none of the information contained in the test dataset is included in the data generation process. If synthetic data is generated based on data in the test dataset, the synthetic data generation algorithms will create new data that is relatively similar to the test data. These new data samples will be incorporated into the training process and may skew the results because the synthetic datasets were generated while knowing what the test dataset looks like [41]. Splitting the datasets before generating new data alleviates this problem.

Second, the purpose of our classifiers is to detect and classify anonymous traffic in the real world. Importantly, our research is not designed to determine classifiers' ability to identify artificially generated anonymous traffic. If we didn't perform a split before generating synthetic data, synthetic data would almost certainly be present in the test dataset after conducting a train test split. Our metrics would be biased as they would reflect the classifiers ability to classify synthetic data making it difficult to predict how the model would perform on real-world traffic.

D. PERFORMANCE METRICS

Accuracy can give an insight into the performance of a classifier, but If the dataset is imbalanced, the model may yield high accuracy on training data and low accuracy in practical application. Therefore, we used four metrics—precision, recall, F1-score, and AUC—to measure model performance. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are the components of these evaluation metrics.

Precision is the proportion of positive classifications that are correct and recall is the percent of TPs that a model predicted. F1-score is calculated from the precision and recall metrics and is used in evaluating the balance between precision and recall. ROC curves represent the TP rate measured against the FP rate. The area under the ROC curve (AUC) and F1 score are metrics which aren't biased by disproportionate data and can better evaluate if a model is overfit.

$$Precision = \frac{TP}{TP + FP * 100} \tag{5}$$

$$Recall = \frac{TP}{TP + FN * 100} \tag{6}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} * 100 \tag{7}$$

E. DATA AUGMENTATION

Previous research in the field of network traffic classification tends to upsample minority classes to the majority class. Although this is a valid methodology, we wanted to experiment with other upsampling and downsampling proportions to get optimal results. In addition to upsampling each class to the majority, eleven new datasets were generated from the Scenario A and Scenario B seed datasets using SMOTE and random undersampling for the sake of maintaining some of their original proportionality.

When trained on our classifiers, The top-performing dataset for Scenario A consisted of 30,000 regular samples, 20,000 VPN samples, and 10,000 Tor samples. The dataset where all classes were upsampled to the majority class (92,659 samples) was the next best performer. For Scenario B, the top performing dataset contained 30,000 samples of each application type. The second best dataset upsampled all eight application types to the majority class which contained 48,020 samples.

With the optimal proportions in mind, SMOTE, VAE, and GANs were employed to generate a new dataset each for Scenario A and another dataset for Scenario B using the aforementioned proportions of 30,000 regular samples, 20,000 VPN samples, and 10,000 Tor samples for Scenario A and 30,000 samples of each application type for Scenario B. In total, four datasets were generated for Scenario A (using SMOTE, VAE, CTGAN, and CopulaGan) and four for Scenario B for a total of eight new datasets. Each dataset was used to train the five classifiers which were then tested on the test dataset to evaluate performance. The results section presents classifiers trained on the 30,000 regular, 20,000 VPN, and 10,000 Tor samples for Scenario A and 30,000 samples for each application type for Scenario B as these were found to be the top performing upsampling strategies.

F. MODEL OPTIMIZATION

The three shallow learning models—Random Forest (RF), XGBoost (XGB), and LightGBM (LGBM)—underwent hyperparameter tuning with grid search. LGBM and XGB were subject to variations in their n_estimators, max_depth, min_child_weight, and eval_metric. For RF, we tuned the n_estimators and max_feature parameters. Each shallow learning model was trained with variations in these parameters and the model with the optimal parameters are presented in the results section.

Hyperparameter tuning was applied to TabNet and the DNN as well. Both models use fastai's built-in lr_find() method. The function returns the optimal learning rate among a valley, slide, steep, or minimum learning rate. After experiments it was found that both the deep learning models perform best around 20 epochs. If more epochs are conducted, the models begin to overfit. The optimal dimension for the DNN was a 15-layer network with 125 nodes in each layer. A batch size of 64 was found to be the optimal.

VII. RESULTS

The following section presents the results and findings of the control, Scenario A, and B experiments. First, we establish how classifiers trained on unbalanced data are maladaptive to diverse data and may overfit to the majority classes. Then, we compare classifier performance in Scenarios A and B using upsampled synthetic data.

A. IMBALANCED VS BALANCED TEST SETS

Figures 5 and 6 depict our classifiers' performance when trained on the imbalanced dataset and tested on balanced and imbalanced test sets for Scenario A and B. This set of experiments was conducted on the CIC-Darknet2020 dataset without any synthetic data. While all classifiers yielded high metrics when tested on the imbalanced data, there was a pronounced and expected drop in F1 and accuracy when evaluated on balanced data. Notably, the deep learning models experienced the greatest performance reduction on the balanced dataset. These results indicate that the anonymous network traffic classifiers tend to overfit due to the skewed nature of training data which may make the classifiers infeasible in real world scenarios. For this reason, all of the following experiments are tested on balanced data and compared to the balanced-tested control unless otherwise stated.

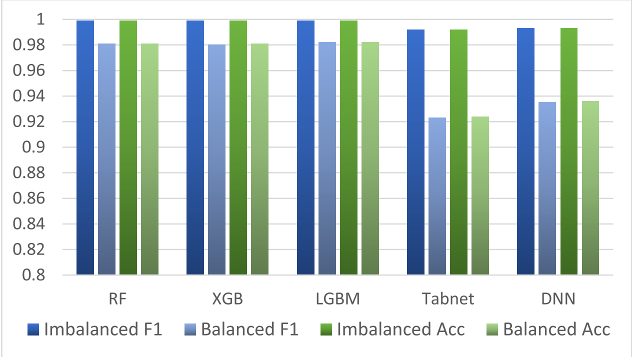

FIGURE 5. Scenario A model comparison when tested on imbalanced and balanced test sets.

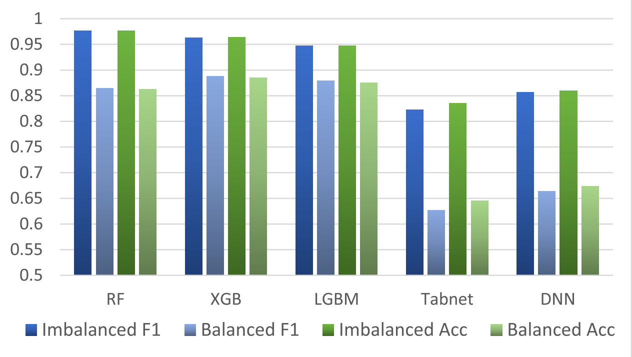

FIGURE 6. Scenario B model comparison when tested on imbalanced and balanced test sets.

B. SCENARIO A RESULTS

All classifiers in Scenario A achieved F1 and AUC scores greater than 90% whether trained on the real data or the synthetically upsampled data. Table 2 contains the results for the Scenario A experiments and is highlighted to accentuate which training data produced the highest metrics for each given model (green) as well as which techniques outperformed the corresponding control model (blue). Each synthetic upsampling technique was used to create a dataset with 30,000 regular, 20,000 VPN, and 10,000 Tor samples as this was found to be optimal. It can be observed that classifiers trained on SMOTE upsampled data had the highest metrics when compared to the other techniques. TabNet and the DNN experienced a greater F1-score improvement from control to SMOTE as compared to the shallow learning models. Furthermore, CTGAN, CopulaGAN, and the VAE saw no major improvement or degradation from the control in this scenario.

Independent of upsampling techniques, the shallow learning classifiers experienced low variability and performed better on average than the deep learning models. This could be attributed to the fact that deep learners tend to require more data than shallow learners, and our seed data may not have been sufficient. Moreover, the deep learners may require

TABLE 2. Results for Scenario A experiments.

	Metric	RF	XGB	LGBM	TabNet	DNN
Control	F1	0.981	0.980	0.982	0.923	0.935
	AUC	0.997	0.998	0.998	0.977	0.985
SMOTE	F1	0.987	0.987	0.989	0.960	0.965
	AUC	0.999	0.999	0.999	0.990	0.994
CTGAN	F1	0.978	0.977	0.978	0.925	0.924
	AUC	0.997	0.998	0.998	0.959	0.972
Copula GAN	F1	0.979	0.981	0.981	0.901	0.920
	AUC	0.998	0.998	0.999	0.963	0.984
VAE	F1	0.977	0.978	0.976	0.931	0.915
	AUC	0.997	0.998	0.998	0.975	0.989

TABLE 3. Results for Scenario B experiments.

	Metric	RF	XGB	LGBM	TabNet	DNN
Control	F1	0.865	0.888	0.879	0.570	0.649
	AUC	0.980	0.992	0.992	0.927	0.934
SMOTE	F1	0.890	0.901	0.903	0.748	0.783
	AUC	0.986	0.993	0.993	0.954	0.959
CTGAN	F1	0.861	0.859	0.840	0.508	0.530
	AUC	0.981	0.987	0.983	0.860	0.840
Copula GAN	F1	0.861	0.861	0.830	0.489	0.494
	AUC	0.981	0.986	0.982	0.832	0.850
VAE	F1	0.852	0.871	0.825	0.569	0.542
	AUC	0.980	0.994	0.981	0.898	0.870

further hyperparameter optimization and training for more epochs to perform on par with the shallow learners.

C. SCENARIO B RESULTS

The results of the Scenario B experiments (Table 3) saw higher deviations in metrics across synthetic techniques and models due to the larger number of class types. For every sampling technique, we generated a dataset with 30,000 samples of each application type. Once again, SMOTE provided the most promising results, improving over control across all classifiers. The shallow classifiers trained on the other synthetic techniques performed similar to the baseline classifiers. On the contrary, our deep learning models performed poorly when compared to the shallow learners and saw large variations across the different techniques. For instance, TabNet's F1 improved by ∼18% from baseline to SMOTE, whereas DNN's F1 degraded by ∼16% from baseline to CopulaGAN.

With the exception of SMOTE, all deep and most shallow learning classifiers degraded in performance when trained on upsampled data. There could be a multitude of causes for this discrepancy. One potential reason could be that the GANs and the VAE are deep learning-based algorithms and may not have been trained for a sufficient number of epochs. Moreover, they may not have had enough seed data to create representative samples. SMOTE doesn't require considerable sample data to produce new samples occupying the same feature space as the original data because it is a statistical technique that does not iteratively learn on sample data.

Figure 7 illustrates confusion matrices and classification results for LGBM when tested on imbalanced, balanced, and upsampled SMOTE data. From these figures, we can see that "browsing" and "email" were the classes with the greatest improvement in F1-scores when upsampled with SMOTE compared to the balanced control results. It should be noted that "browsing" had the second largest number of original samples while "email" contained the least. This implies that the model was biased towards browsing traffic and had a large number of false positives. After training on SMOTE upsampled data the F1-scores for browsing and email classification improved by 5.7% and 8% respectively. When evaluated against the balance control, all application types improved in F1-score with the exception of "chat" which didn't see any variation in the result.

TABLE 4. Result Comparison with CIC-Darknet2020 Research.

Paper	Top Classifier	Synthetic Data	Traffic Type	Application Type
Gupta [21]	XGB	No	98% acc. (3 classes)	N/A
Iliadis [22]	RF	No	98.7% acc. (4 classes)	N/A
Lashkari [11]	2D CNN	No	N/A	86% acc. & f1 (8 classes)
Allhusen [23]	RF	No	99.5% acc. (2 classes)	N/A
Balanced Results	LGBM	No	98.2% F1 (3 classes)	88.8% F1 (8 classes)
Synthetic Results	LGBM	Yes (4)	98.7% F1 (3 classes; SMOTE)	90.3% F1 (8 classes; SMOTE)

D. SUMMARY

Across both scenarios, SMOTE was the top performing generative technique, improving classifier metrics over the control for every classifier. The other balancing techniques performed near baseline for shallow learners, but with additional optimization there may be further improvements. Moreover, we showed that non-SMOTE generative techniques can degrade performance, especially in deep learning classifiers. Table 4 contextualizes our results with prior research. Our traffic results perform on par with prior research while our application results showed improvement 4% improvement over Lashkari et al.'s [9] application type experiments. Furthermore, we measure model performance using F1-score as it is less susceptible to overfitting and our results show that synthetic data is viable in this domain. Compared to prior studies, by exploring multiple generative techniques to address the imbalanced classes, we are able to optimize model performance.

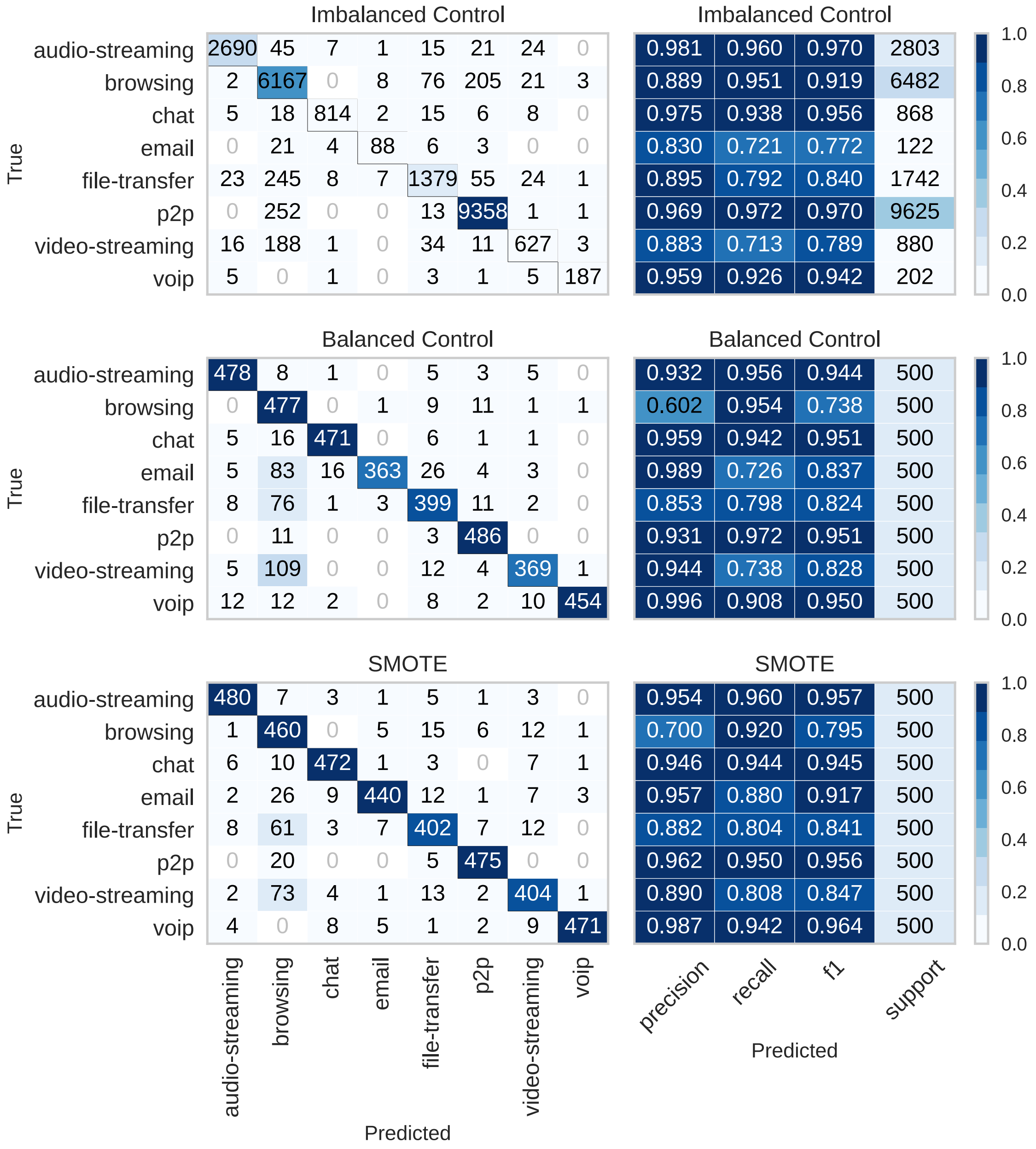

FIGURE 7. Scenario B confusion matrices and classification results for LGBM.

VIII. LIMITATIONS AND FUTURE WORK

As the scope of this work is to assess the efficacy of synthetic data in the field of anonymous traffic and application categorization, it does not optimize every step in the experimental workflow. Future works may benefit from implementing additional SMOTE variants (such as ADASYN) while training the CopulaGAN, CTGAN, and VAE for more epochs. Furthermore, the models observed in this study could be refined through greater hyperparameter tuning and by conducting more exhaustive grid searches on the shallow learning classifiers. This work used SMOTE as a baseline to generate 12 new datasets of varying proportions for each class. Testing further upsampling ratios with other synthetic data generation techniques as the guiding heuristic may improve overall performance. Also, additional metrics such as synthetic data generation time of the discussed methods may be an important consideration during real world implementation of these classifiers. Future work could use multi-criterion decision making to evaluate the proposed methods based on factors other than accuracy and F1.

While the CIC-Darknet2020 dataset contains a large variety of anonymous traffic and application types, there are still several encryption and anonymity protocols and applications such as HTTPS, SSH, and SSL/TLS that are not included within this work. Characterization of those untested types may be necessary for an organization looking to deploy similar models. Moreover, the literature is lacking experimentations on the optimal flow interval, so further experimentation is encouraged to regenerate the tabular dataset from the raw pcap traffic data over different flow intervals.

The experiments would benefit from larger amounts of real data because TabNet and the DNN were likely impacted more than the shallow learners from the limited number of samples in the seed and test datasets. Furthermore, testing our models with balanced data resulted in a test set with ∼600 samples for each application type, which may not fully encapsulate the variety of real-world data.

The CMU-SynTraffic-2022 dataset provides a multitude of avenues for future research. The dataset is more robust compared to many available anonymous traffic datasets and may be incorporated in research works to characterize anonymous traffic and application types. Additionally, researchers may use the dataset to judge how their generative model compares to the generative techniques synthesized to create samples in the CMU-SynTraffic-2022 dataset, or the samples for each generative technique could be analyzed to determine the impact generative models have on classifiers' performance. CMU-SynTraffic-2022 provides a means to analyze synthetic data and its source or algorithm. For instance, it may be the case that samples generated by one generative technique could cluster with samples from another generative technique.

IX. CONCLUSION

In this work, we analyzed the performance of RF, XGB, LGBM, a DNN, and TabNet on a variety of synthetic data generation techniques. First, the classifiers were trained on the imbalanced CIC-Darknet2020 dataset and tasked with classifying samples from imbalanced and balanced test sets. It was demonstrated that, in this experiment, the models experienced performance degradation and struggled to classify minority classes because they were trained on skewed data. Next, four additional datasets were generated using a CTGAN, CopulaGAN, VAE, and SMOTE and the classifiers were retrained on these datasets to evaluate how each technique alleviates the imbalanced problem. Ultimately, the additional datasets were amalgamated into a complete dataset dubbed CMU-SynTraffic-2022 and open sourced for future synthetic and network traffic research [42], [43].

In our two-phased experiments, Scenario A classified Tor, VPN, and regular traffic while Scenario B aimed to differentiate among eight anonymous application types. In both scenarios, SMOTE consistently provided better metrics than both the control set and the other synthetic datasets. Scenario B also showed that deep learning classifiers are impacted more than shallow learning classifiers when upsampling with synthetic data. Our shallow boosting and bagging algorithms (XGB and LGBM) were the top performers among the classifiers.

Due to the inherently imbalanced nature of anonymous network traffic, machine and deep learning classifiers often experience severe performance degradation in real-world applications. Our work demonstrated the viability of synthetic data in the domain of anonymous network traffic classification through a comprehensive comparison of data generation techniques. Furthermore, we addressed the knowledge gap in existing research by directly comparing several generative techniques and their capability to represent real network data. By testing techniques currently unused in this domain (VAE, GAN Variants) and observing how different classifier types perform on synthetic data, researchers may better implement the best generative technique for network traffic tasks. Through the creation of the CMU-SynTraffic-2022 dataset, we provide a means for further network traffic and synthetic data research. Finally, the proposed methodology presented in this work could be translated to other domains with heavily imbalanced data to potentially improve model performance.

ACKNOWLEDGMENT

Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding sources. The authors would like to thank Jackson Warren and Karson Fye for their early help with getting this research project started.

REFERENCES

[1] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, "Denoising adversarial autoencoder for obfuscated traffic detection and recovery," in *Machine Learning for Networking*. Cham, Switzerland: Springer, 2020, pp. 99–116, doi: 10.1007/978-3-030-45778-5_8.

[2] M. Kim and A. Anpalagan, "Tor traffic classification from raw packet header using convolutional neural network," in *Proc. 1st IEEE Int. Conf. Knowl. Innov. Invention (ICKII)*, Jul. 2018, pp. 187–190, doi: 10.1109/ICKII.2018.8569113.

[3] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Proc. IEEE Int. Conf. Intell. Secur. Inform. (ISI)*, Jul. 2017, pp. 43–48, doi: 10.1109/ISI.2017.8004872.

[4] S. E. Gómez, L. Hernández-Callejo, B. C. Martínez, and A. J. Sánchez-Esguevillas, "Exploratory study on class imbalance and solutions for network traffic classification," *Neurocomputing*, vol. 343, pp. 100–119, May 2019. 10.1016/j.neucom.2018.07.091.

[5] Y. Guo, G. Xiong, Z. Li, J. Shi, M. Cui, and G. Gou, "Combating imbalance in network traffic classification using GAN based oversampling," in *Proc. IFIP Netw. Conf. (IFIP Networking)*, Jun. 2021, pp. 1–9, doi: 10.23919/IFIPNetworking52078.2021.9472777.

[6] 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, no. 1, pp. 321–357, Jan. 2002, doi: 10.1613/jair.953.

[7] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[9] P. Wang, S. Li, F. Ye, Z. Wang, and M. Zhang, "PacketCGAN: Exploratory study of class imbalance for encrypted traffic classification using CGAN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7, doi: 10.1109/ICC40277.2020.9148946.

[10] N. Jadav, N. Dutta, H. K. D. Sarma, E. Pricop, and S. Tanwar, "A machine learning approach to classify network traffic," in *Proc. 13th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jul. 2021, pp. 1–6, doi: 10.1109/ECAI52376.2021.9515039.

[11] A. Habibi Lashkari, G. Kaur, and A. Rahali, "DIDarkNet: A contemporary approach to detect and characterize the darknet traffic using deep image learning," in *Proc. 10th Int. Conf. Commun. Netw. Secur.*, Nov. 2020, pp. 1–13, doi: 10.1145/3442520.3442521.

[12] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Privacy*, 2016, pp. 407–414, doi: 10.5220/0005740704070414.

[13] J. A. Caicedo-Muñoz, A. L. Espino, J. C. Corrales, and A. Rendón, "QoS-classifier for VPN and non-VPN traffic based on time-related features," *Comput. Netw.*, vol. 144, pp. 271–279, Oct. 2018, doi: 10.1016/j.comnet.2018.08.008.

[14] S. Miller, K. Curran, and T. Lunney, "Multilayer perceptron neural network for detection of encrypted VPN network traffic," in *Proc. Int. Conf. Cyber Situational Awareness, Data Analytics Assessment (Cyber SA)*, Jun. 2018, pp. 1–8, doi: 10.1109/CyberSA.2018.8551395.

[15] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy*, 2017, pp. 1–10, doi: 10.5220/0006105602530262.

[16] Y. Huo, H. Ge, L. Jiao, B. Gao, and Y. Yang, "Encrypted traffic identification method based on multi-scale spatiotemporal feature fusion model with attention mechanism," *Proc. 11th Int. Conf. Comput. Eng. Networks.* Singapore: Springer, Nov. 2021, pp. 857–866, doi: 10.1007/978-981-16-6554-7_92.

[17] A. Gurunarayanan, A. Agrawal, A. Bhatia, and D. K. Vishwakarma, "Improving the performance of machine learning algorithms for TOR detection," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2021, pp. 439–444, doi: 10.1109/ICOIN50884.2021.9333989.

[18] Z. Wang, P. Wang, X. Zhou, S. Li, and M. Zhang, "FLOWGAN: Unbalanced network encrypted traffic identification method based on GAN," in *Proc. IEEE Int. Conf Parallel Distrib. Process. Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, Dec. 2019, pp. 975–983, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00141.

[19] Z. Okonkwo, E. Foo, Q. Li, and Z. Hou, "A CNN based encrypted network traffic classifier," in *Proc. Australas. Comput. Sci. Week*. New York, NY, USA: Association for Computing Machinery, Feb. 2022, doi: 10.1145/3511616.3513101.

[20] J. Li, D. Wang, S. Li, M. Zhang, C. Song, and X. Chen, "Deep learning based adaptive sequential data augmentation technique for the optical network traffic synthesis," *Opt. Exp.*, vol. 27, no. 13, p. 18831, Jun. 2019, doi: 10.1364/OE.27.018831.

[21] N. Gupta, V. Jindal, and P. Bedi, "Encrypted traffic classification using extreme gradient boosting algorithm," in *Advances in Intelligent Systems and Computing*. Singapore: Springer, Aug. 2021, pp. 225–232, doi: 10.1007/978-981-16-3071-2_20.

[22] L. A. Iliadis and T. Kaifas, "DarkNet traffic classification using machine learning techniques," in *Proc. 10th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, Jul. 2021, pp. 1–4, doi: 10.1109/MOCAST52088.2021.9493386.

[23] A. Al-Omari, A. Allhusen, A. Wahbeh, M. Al-Ramahi and I. Alsmadi, "Dark web analytics: A comparative study of feature selection and prediction algorithms," in *Proc. Int. Conf. Intell. Data Sci. Technol. Appl. (IDSTA)*, 2022, pp. 170–175, doi: 10.1109/IDSTA55301.2022.9923042.

[24] S. Arik and T. Pfister, "TabNet: Attentive interpretable tabular learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 8, May 2021, pp. 6679–6687, doi: 10.1609/aaai.v35i8.16826.

[25] B. Sun, L. Yang, W. Zhang, M. Lin, P. Dong, C. Young, and J. Dong, "SuperTML: Two-dimensional word embedding for the precognition on structured tabular data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–9, doi: 10.1109/CVPRW.2019.00360.

[26] G. Ajay and P. Krishnan. (2018). *A Study and Analysis of Effective Data transmission Using UDP*. [Online]. Available: https://www.ijser.org/researchpaper/A-Study-and-Analysis-of-Effective-Data-transmission-Using-UDP.pdf

[27] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 3, pp. 261–277, 2001, doi: 10.1023/a:1017934522171.

[28] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009. [Online]. Available: https://dl.acm.org/doi/10.5555/2976248.2976433

[29] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1322–1328, doi: 10.1109/IJCNN.2008.4633969.

[30] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," 2018, *arXiv:1806.03384*.

[31] L. Xu and K. Veeramachaneni, "Synthesizing tabular data using generative adversarial networks," 2018, *arXiv:1811.11264*.

[32] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," 2019, *arXiv: 1907.00503*.

[33] *CopulaGAN Model*. Accessed: Feb. 12, 2022. [Online]. Available: https://sdv.dev/SDV/user_guides/single_table/copulagan.html

[34] S. Kamthe, S. Assefa, and M. Deisenroth, "Copula flows for synthetic data generation," 2021, *arXiv:2101.00598*.

[35] S. Bourou, A. El Saer, T.-H. Velivassaki, A. Voulkidis, and T. Zahariadis, "A review of tabular data synthesis using GANs on an IDS dataset," *Information*, vol. 12, no. 9, p. 375, Sep. 2021, doi: 10.3390/info12090375.

[36] A. Singh and T. Ogunfunmi, "An overview of variational autoencoders for source separation, finance, and bio-signal applications," *Entropy*, vol. 24, no. 1, p. 55, Dec. 2021, doi: 10.3390/e24010055.

[37] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2016, pp. 399–410, doi: 10.1109/DSAA.2016.49.

[38] Y.-T. Chen and M. C. Chen, "Using chi-square statistics to measure similarities for text categorization," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3085–3090, 2011, doi: 10.1016/j.eswa.2010.08.100.

[39] Z. Drezner, O. Turel, and D. Zerom, "A modified Kolmogorov–Smirnov test for normality," *Commun. Statist. Simul. Comput.*, vol. 39, no. 4, pp. 693–704, Mar. 2010, doi: 10.1080/03610911003615816.

[40] D. Westreich, J. Lessler, and M. J. Funk, "Propensity score estimation: Neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression," *J. Clin. Epidemiol.*, vol. 63, no. 8, pp. 826–833, Aug. 2010, doi: 10.1016/j.jclinepi.2009.11.020.

[41] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data," *BMC Bioinf.*, vol. 14, no. 1, pp. 1–16, Mar. 2013, doi: 10.1186/1471-2105-14-106.

[42] D. Cullen, J. Halladay, N. Briner, and R. Basnet. (2022). *CMU-SynTraffic-2022*. IEEE DataPort. [Online]. Available: https://ieee-dataport.org/documents/cmu-syntraffic-2022

[43] J. Halladay, D. Cullen, and N. Briner. (Jun. 2022). *CMUDarknet*. Github Repository. [Online]. https://github.com/Colorado-Mesa-University-Cybersecurity/CMU_Darknet_Research

DRAKE CULLEN is currently pursuing the bachelor's degree in computer science with minors in cybersecurity and mathematics with Colorado Mesa University (CMU). He is also the Former President of the Cybersecurity Club, the President of Upsilon Pi Epsilon (the International Honor Society for Computing and Informatics), and the Treasurer of Kappa Mu Epsilon (the National Mathematics Honor Society) at CMU.

JAMES HALLADAY is currently pursuing the bachelor's degree in math and computer science with Colorado Mesa University. His research interest includes application of machine learning to the domain of cyber security.

NATHAN BRINER is currently pursuing the bachelor's degree in computer science and a minor in mathematics with Colorado Mesa University. He is also the Vice President of CMU's Computer Science Club and the Vice President of Upsilon Pi Epsilon (the International Honor Society for Computing and Informatics) at CMU.

RAM BASNET received the B.S. degree in computer science from Colorado Mesa University (CMU), in 2004, and the M.S. and Ph.D. degrees in computer science from New Mexico Tech, in 2008 and 2012, respectively. He is currently a Professor of computer science and cybersecurity with CMU. His research interests include the areas of information assurance, machine learning, and computer science pedagogy.

JEREMY BERGEN received the M.S. degree in computer science from the Georgia Institute of Technology, in 2021, where his focus was on machine learning. He is currently an Assistant Professor of computer science at Colorado Mesa University. His current research interests include computer vision, gamification in education, and cyber security research.

TENZIN DOLECK received the Ph.D. degree from McGill University, in 2017. He is currently working as a Canada Research Chair and an Assistant Professor with Simon Fraser University.