

Original Research

Received 09 March 2026

Revised 12 April 2026

Accepted 25 May 2026

Natural Resources for Human
Health 2026, Vol. 6 (17s): 849–866

KEYWORDS:

Class imbalance; synthetic data
generation; generative adversarial
network; variational autoencoder;

Hybrid VAE-GAN Generative Oversampling for Extreme Class Imbalance in Insider Threat Detection

Mahesh Kumar Anem¹, Abdul Ahad^{2*}

¹School of Engineering, Anurag University, Hyderabad 500088, Telangana, India; ORCID id (0009-0002-4763-5161)

²School of Engineering, Anurag University, Hyderabad 500088, Telangana, India; ORCID id (0000-0002-1282-2189)

*Corresponding author: Abdul Ahad, School of Engineering, Anurag University, Venkatapur, Ghatkesar, Medchal-Malkajgiri District, Hyderabad 500088, Telangana, India.

10112. Email: ahadbabu@gmail.com

ABSTRACT: Insider threat detection works on data where malicious days are very rare. In the CERT r4.2 corpus used here, 1,364 malicious user-days stand against 328,921 benign ones, a ratio of 241 to 1. Interpolation based oversamplers such as SMOTE copy rather than create, and treat count data as if it were continuous. This paper presents Hybrid Generative Oversampling. A conditional variational autoencoder and a conditional Wasserstein generative adversarial network are fitted on the minority rows of each training fold, samples are drawn from both, and the pool is then reduced by a boundary test and a novelty test. Every column receives the head that suits it: a softmax over observed counts, a gate for zero inflated columns, a value head for the rest. Under employee disjoint five fold cross validation, with synthetic rows confined to the training folds, the method raises the Random Forest F1 score from 0.369 to 0.409 and the Matthews correlation from 0.392 to 0.440, and it improves on SMOTE in precision recall area in five folds out of five. Its rows also lie 1.82 times the real spacing away from the closest real record, against 0.32 for SMOTE.

1. INTRODUCTION

An insider is an employee, a contractor or a partner who already holds valid credentials. When such a person turns hostile, there is no broken lock and no failed login to look at. The person simply does slightly more of what is already allowed, copies a few more files, sends a few more attachments outside, and stays a little longer in the evening. Surveys of the field agree that this is what makes the problem hard, and that the detection task is really a search for a small change in an otherwise normal pattern (Homoliak et al., 2019; Yuan and Wu, 2021).

The statistical shape of the problem is severe. In the CERT r4.2 corpus, which is the standard public benchmark for this work (Glasser and Lindauer, 2013), one thousand employees are observed for about seventeen months. After the raw logs are aggregated into one row per employee per day, in the manner that is now usual for this data (Le et al., 2020), there are 330,285 user-days, of which only 1,364 are part of a malicious act. The imbalance ratio is 241 to 1 and the positive rate is 0.41 percent. A classifier that answers benign for every row is correct 99.59 percent of the time and is completely useless.

The standard reply to imbalance is to resample the training set. SMOTE and the methods built on it create new minority rows by interpolating between a real minority row and one of its minority neighbours (Chawla et al., 2002; Han et al., 2005; He et al., 2008). These methods are simple, fast and well studied (Fernández et al., 2018), but they carry two assumptions that do not hold for behavioural logs. The first assumption is that the straight line between two minority rows lies inside the minority region. When the minority rows come from different attack scenarios, or from employees who behave differently, the midpoint can be a day that no employee has ever had. The second assumption is that the features are continuous. In our matrix, twenty five of the thirty nine columns take fewer than fifty distinct values, and several columns are zero on more than ninety percent of the days. Interpolating two integer counts gives a fraction, and a fraction of a

universal serial bus insertion has no meaning.

Deep generative models offer a different route. A generative adversarial network (Goodfellow et al., 2014) or a variational autoencoder (Kingma and Welling, 2014) learns a distribution and then draws from it, so the new rows need not lie on a line between two old ones. Conditional variants (Mirza and Osindero, 2014) allow the attack type to be requested by name. Work on tabular generation has shown that the discrete columns must be handled as discrete (Xu et al., 2019), and oversampling with conditional Wasserstein networks has already been reported as competitive on credit and marketing data (Engelmann and Lessmann, 2021). What has not been settled is whether this holds on a security log with an imbalance of more than two hundred to one, and whether the extra cost is repaid.

This paper answers that question with a method and a measurement. The contributions are the following.

- A generator representation in which every column receives the head that matches it. A count column with a small support gets a softmax over the values that were actually observed, a continuous column that is often zero gets a Bernoulli gate and a value head, and the remaining columns get a value head alone. The count heads stay differentiable through a Gumbel softmax relaxation (Jang et al., 2017).
- Hybrid Generative Oversampling, written HGO. A conditional variational autoencoder and a conditional Wasserstein generative adversarial network with a gradient penalty (Arjovsky et al., 2017; Gulrajani et al., 2017) are fitted on the same minority rows, and a candidate pool is drawn from both. A boundary filter removes the candidates that fall closer to the benign manifold than to the malicious one, and a novelty filter removes the candidates that land almost on top of a real record.
- Scenario conditioned sampling with a square root reweighting, so that the rare attack scenarios are not represented in proportion to how rare they are.
- An evaluation that keeps every synthetic row inside the training folds, splits the folds by employee rather than by row, and reports both detection quality and the quality of the synthetic data, including a distance check that shows how far the generated rows sit from real employee records.
- An honest account of where the method helps and where it does not. The gains appear with a Random Forest detector and at a small oversampling ratio. With a linear model or a small neural network on this data, no oversampler beat the plain imbalanced baseline in precision recall area.

The rest of the paper is organised as follows. Section 2 reviews the related work. Section 3 states the problem. Section 4 describes the proposed method. Section 5 gives the experimental setup, Section 6 reports and discusses the results, and Section 7 concludes.

2. RELATED WORK

2.1 Resampling for Imbalanced Learning

The imbalanced learning literature is large and well summarised (He and Garcia, 2009; Krawczyk, 2016). The simplest correction is random oversampling, which repeats minority rows until the classes are closer in size. It costs nothing, but it creates exact duplicates, which pushes a flexible model towards memorising those rows. SMOTE (Chawla et al., 2002) replaced duplication with interpolation, and the idea was extended in many directions. Borderline SMOTE (Han et al., 2005) restricts the interpolation to minority rows that already sit near the decision boundary, on the argument that these are the informative ones. ADASYN (He et al., 2008) goes further and generates more rows around the minority points that are hardest to classify. A fifteen year review (Fernández et al., 2018) found that no single member of the family is best everywhere, and that the gain depends strongly on the classifier that consumes the data. Cost sensitive learning is the alternative that changes the loss instead of the data, and it is included here as a baseline.

2.2 Deep Generative Models for Tabular Data

Generative adversarial networks (Goodfellow et al., 2014) learn by putting a generator against a discriminator. The original formulation is difficult to train, and the Wasserstein formulation (Arjovsky et al., 2017) with a gradient penalty (Gulrajani et al., 2017) is now the practical choice, particularly when the training set is small, because it keeps giving the generator a usable gradient after the critic has become strong. Variational autoencoders (Kingma and Welling, 2014) take a different path

and learn an explicit latent distribution, which makes them stable but tends to produce smoother, more averaged output. Conditioning (Mirza and Osindero, 2014) lets the caller ask for a particular class or subtype.

Tabular data needed its own treatment. CTGAN and its autoencoder counterpart (Xu et al., 2019) showed that mixed columns must be modelled as mixed, with a mode specific normalisation for the continuous columns and a categorical representation for the discrete ones, and that a Gumbel softmax (Jang et al., 2017) keeps the categorical part trainable. The representation used in this paper follows that principle but is simpler, because the columns here are counts with a known support rather than free categorical labels.

2.3 Generative Oversampling and Security Data

Douzas and Bacao (2018) were among the first to use a conditional generative adversarial network as an oversampler and reported gains over SMOTE on a range of benchmark sets. Engelmann and Lessmann (2021) refined this with a conditional Wasserstein network that handles numerical and categorical columns and adds an auxiliary classifier loss, and reported competitive results on seven real datasets. Sharma et al. (2022) combined the two ideas in the other order, using SMOTE output as the starting point for the generator, which helps when the minority set is too small for a generator to learn from directly.

On the security side, most insider threat work has concentrated on the detector rather than on the data. Tuor et al. (2017) used an unsupervised recurrent model over structured log streams. Le et al. (2020) studied how the choice of aggregation window changes what a machine learning model can see, and observed that the extreme imbalance is one of the main obstacles. Reviews of deep learning for this problem (Yuan and Wu, 2021) list class imbalance as an open issue but report few controlled comparisons of the remedies. The gap this paper fills is a like for like comparison, on one security corpus, under one protocol, of the classical resamplers against generative ones, with the quality of the generated rows measured rather than assumed. **Table 1** places this work beside the studies that are closest to it.

Table 1. Position of this work against representative earlier work.

Work	Data	Generator	Discrete columns	Filtering of samples	Imbalance ratio
Chawla et al. (2002)	UCI benchmarks	interpolation	not treated	none	up to 20 to 1
Han et al. (2005)	UCI benchmarks	interpolation near boundary	not treated	boundary selection of seeds	up to 20 to 1
He et al. (2008)	UCI benchmarks	interpolation, density weighted	not treated	none	up to 30 to 1
Douzas and Bacao (2018)	71 benchmark sets	conditional GAN	not treated	none	up to 130 to 1
Xu et al. (2019)	tabular benchmarks	conditional GAN and VAE	mode specific and one hot	none	not an imbalance study
Engelmann and Lessmann (2021)	7 tabular sets	conditional WGAN	one hot with auxiliary loss	none	up to 50 to 1
Sharma et al. (2022)	9 benchmark sets	SMOTE then GAN	not treated	none	up to 130 to 1
Tuor et al. (2017)	CERT r6.2	no generator	not applicable	not applicable	about 1000 to 1
Le et al. (2020)	CERT r4.2 and r6.2	no generator	not applicable	not applicable	about 200 to 1
This work	CERT r4.2	conditional VAE and WGAN-GP together	softmax and gate heads	boundary and novelty filters	241 to 1

3. PROBLEM FORMULATION

Let the organisation employ users $U = \{u_1, \dots, u_N\}$ observed over calendar days $T = \{t_1, \dots, t_M\}$. For a user u and a day t the monitoring system produces a feature vector $x(u, t)$ in $\mathbb{R}^d$ that summarises the authentication, removable media, file, electronic mail and web activity of that person on that day. The label $y(u, t)$ is one when the ground truth marks the day as part of a malicious act and zero otherwise. The observed set is $S = \{(x_i, y_i, s_i)\}$, $i = 1, \dots, n$, where s_i is the attack scenario of a malicious day and is undefined for a benign day.

Write the minority set as $S^+ = \{x_i : y_i = 1\}$ with size n^+ and the majority set as S^- with size n^- . The imbalance ratio is $IR = n^- / n^+$.

On this corpus $n^+ = 1,364$, $n^- = 328,921$ and $IR = 241.1$. The task is to learn a scoring function f that ranks malicious user-days above benign ones, using a training set that has been enlarged by a set of synthetic minority rows.

An oversampler is a map G that takes the training minority S^+ and a requested count k and returns k synthetic rows. We ask three things of the rows it returns. They should be plausible, meaning that the value of each column and the relations between columns should match what real malicious days look like. They should be new, meaning that they should not be copies or near copies of the rows the generator was shown, both because copies teach the classifier nothing and because they would carry a real employee's day into any shared artefact. Finally they should be useful, meaning that a classifier trained on the enlarged set should rank real held out malicious days better than the same classifier trained on the original set. These three requirements are measured separately in Section 6, because a method can satisfy one and fail another, and the results below show exactly that.

The count k is fixed by a target ratio. Let q be the desired size of the synthetic minority as a share of the majority count, so $k = \text{round}(q \times n^-)$. Every method in this study is asked for the same k , which means that a difference between two rows of a results table comes from how the rows were made and not from how many were made. **Table 2** collects the notation that is used from here on.

Table 2. Notation used in the paper.

Symbol	Meaning
$x(u, t)$	feature vector of user u on day t , $d = 39$ columns
$y(u, t)$	label, one when the day is part of a malicious act
s	attack scenario of a malicious day, one of three
S^+, S^-	minority and majority sets of a training
fold n^+, n^-	sizes of S^+ and S^-
IR	imbalance ratio, n^- divided by n^+
q	target size of the synthetic minority as a share of n^-
k	number of synthetic rows requested, $\text{round}(q \times n^-)$
z	latent vector of a generator
C	codec that maps a user-day to the detector space and back
V	mixed transform used inside the generators
$d_{\min}, d_{\max}$	distance to the x -th nearest real minority or majority row
δ	novelty threshold, 5th percentile of real minority spacing
α	share of the candidate pool drawn from the autoencoder

4. PROPOSED METHOD

4.1 Overview

Figure 1 shows the whole pipeline. The user-day matrix is split by employee, so that no employee appears in more than one of the training, validation and test parts. A codec is fitted on the training rows and maps each column through its own empirical quantile function to a standard normal variable. This is the space in which the detectors are trained and in which the filters measure distance. The malicious rows of the training part, and only those rows, are then passed to the generative stage. The stage produces k synthetic rows, these are added to the real training rows, a detector is fitted on the result, and the detector is scored on the held out employees, whose rows are all real.

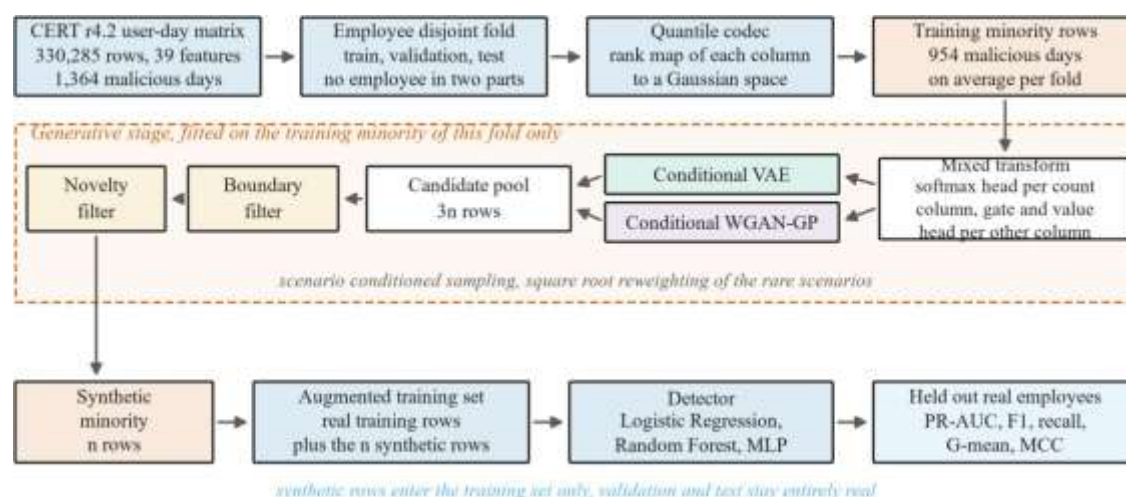


Figure 1. The Hybrid Generative Oversampling pipeline. The generative stage sees the malicious rows of the training employees and nothing else.

Two design decisions in this picture deserve to be stated early. The first is that the split is by employee. If the split were by row, the same employee would appear on both sides, and since an insider repeats a pattern over many days, the model would be recognising the person rather than the behaviour. The second is that the generative stage is inside the fold. A generator fitted on all malicious rows and then used in every fold would have seen the test employees, and the reported numbers would be optimistic.

4.2 Detector Space and Generator Space

Two different representations are used, because the detector and the generator want different things.

The detector space is produced by a quantile codec. Each column is mapped through its own empirical quantile function, estimated on the training rows, to a standard normal variable, and the inverse map returns a value on the original scale. Integer columns are rounded on the way back and every column is clipped to the range seen in training, so no synthetic row can carry a negative count or a value larger than anything ever observed. This keeps the linear and neural detectors on a sensible scale and gives the distance based filters a space in which one unit means the same thing in every direction.

The generator space is different. When each of the thirty nine columns is set against how many distinct values it takes and how often it is zero, as shown for the whole corpus in Section 5.1, the columns fall into three groups. Twenty five of them take fewer than fifty distinct values and are plainly counts, for example the number of logons or the number of removable media insertions. Six of the remaining columns are continuous but are zero on a large share of days, for example the amount of web browsing done after office hours. The last eight are continuous and almost always non zero, for example the hour at which the first session of the day began. A single squared error head over all thirty nine columns has to answer with an average, and the average of a spike at zero and a spike at one is one half, a value that never occurs. The mixed transform therefore gives each group its own head:

- a count column becomes a softmax over the values that were actually observed in the training minority, so the generator chooses a value rather than approximating one,
- a zero inflated continuous column becomes a Bernoulli gate together with a value head, and the value head is trained only on the days where the column is not zero,

- a plain continuous column becomes a value head alone, on the quantile scale of its own non zero part.

For this corpus the transform produces a vector of 475 entries on average, against the 39 raw columns. The inverse draws a value from each softmax, draws each gate, inverts the quantile map for the value heads and applies the gates, which returns a user-day that is valid by construction.

4.3 Conditional Variational Autoencoder

The first generator is a conditional variational autoencoder, drawn in panel (a) of **Figure 2**. The encoder takes the mixed encoding of a user-day together with the one hot attack scenario and produces a mean and a log variance over a latent vector of size 24. The decoder takes a latent vector and the same scenario code and produces the three groups of heads. The training objective is the sum of a reconstruction term and the Kullback Leibler term of the standard formulation (Kingma and Welling, 2014), where the reconstruction term is a cross entropy on each count head, a binary cross entropy on each gate and a squared error on each value head, the last one masked so that a gated off column contributes nothing:

$$\mathcal{L} = \sum_c \text{CE}(\text{head}_c, x_c) + \sum_g \text{BCE}(\text{gate}_g, x_g) + \sum_v m_v (\text{head}_v - x_v)^2 + \beta \text{KL}(q(z | x, s) \parallel p(z)) \quad (1)$$

The weight β is set to 0.5 and is warmed up linearly over the first thirty percent of the epochs, which stops the latent space from collapsing before the decoder has learned anything. Training runs for 600 epochs with the Adam optimiser at a learning rate of 0.001 and a batch of 128. Because the minority set of a fold holds fewer than a thousand rows, one epoch is only eight updates and the whole fit is quick.

4.4 Conditional Wasserstein Network

The second generator, in panel (b) of **Figure 2**, is a conditional Wasserstein generative adversarial network with a gradient penalty. The generator maps a noise vector of size 32 and the scenario code to the same mixed layout. The critic reads a user-day and the scenario code and returns a real number. The critic loss is the usual Wasserstein estimate with the penalty of Gulrajani et al. (2017):

$$L_D = \mathbb{E}[D(x_{\text{fake}}, s)] - \mathbb{E}[D(x_{\text{real}}, s)] + \lambda \mathbb{E}[(\|\nabla D(x_{\text{mix}}, s)\| - 1)^2] \quad (2)$$

with λ set to 10, five critic steps for each generator step, and Adam at a learning rate of 0.0002 with momentum terms 0.5 and 0.9. Weight clipping was not used because on a minority set of this size it drives the critic to a nearly constant function and the generator stops improving. The count heads are sampled with a Gumbel softmax at temperature 0.2 (Jang et al., 2017) and the gates are passed through a sigmoid, so the critic receives a differentiable relaxation of a discrete choice and the gradient reaches the generator.

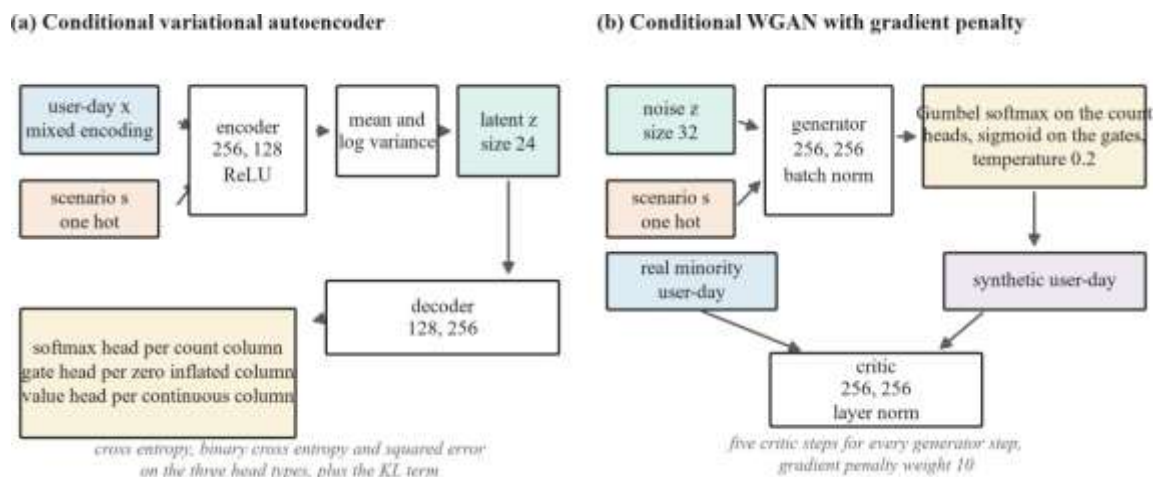


Figure 2. The two generators. Both write into the same mixed layout, so their output can be pooled without any conversion.

4.5 Hybrid Sampling and the Two Filters

The two generators fail in different ways, and Section 6 measures this. The autoencoder keeps the relations between columns better but smooths the individual columns, while the adversarial network reproduces each column very well but

loses some of the joint structure. Rather than choosing, HGO draws a share α of the candidate pool from the autoencoder and the remainder from the adversarial network, with α set to 0.5, and then removes the weak candidates with two filters (**Figure 3**). The pool is three times the number of rows that will finally be kept.

The boundary filter asks whether a candidate sits in a part of the space that malicious days occupy. For a candidate z it computes $d_{\min}$, the distance to its fifth nearest real minority row, and $d_{\max}$, the distance to its fifth nearest real majority row, where the majority reference set is drawn at random and cut to the same size as the minority set so that the two distances are measured at the same sampling density. The candidate is kept when $d_{\min}$ is not greater than $d_{\max}$. A candidate that fails this test would place a positive label inside a region full of benign days, which is exactly the failure that interpolation between distant minority rows produces.

The novelty filter asks the opposite question. Let δ be the fifth percentile of the distances between a real minority row and its own nearest real minority neighbour. A candidate closer than δ to some real minority row is discarded. This removes rows that are effectively copies. The reason is partly statistical, since a copy adds no information and encourages memorisation, and partly practical, because a synthetic corpus that reproduces a real employee's day is not safe to share.

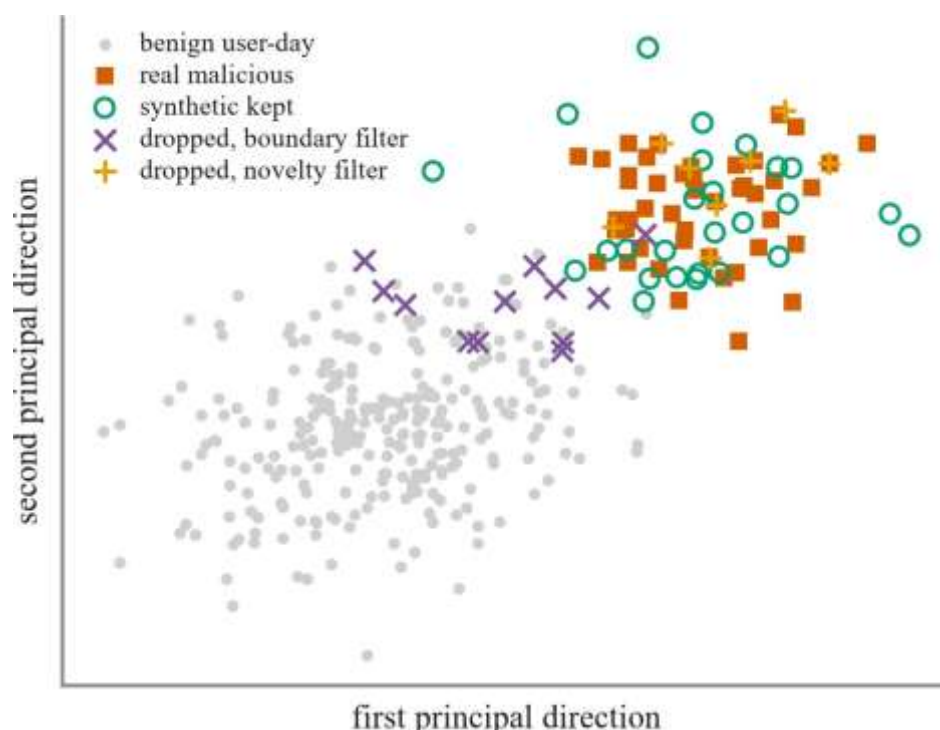


Figure 3. What the two filters remove. Candidates inside the benign region fail the boundary test and candidates sitting on a real record fail the novelty test.

The scenario codes for the pool are not drawn in proportion to the observed scenario counts. On this corpus the three scenarios hold 156, 1,188 and 20 malicious user-days, so proportional sampling would give the third scenario about one row in seventy. The counts are therefore reweighted by their square root before the pool is allocated, which lifts the rare scenarios without letting them dominate.

4.6 The Complete Procedure

Algorithm 1 Hybrid Generative Oversampling for one fold

Input: training rows X , labels y , scenarios s , ratio q , share α ,
neighbourhood size κ , pool factor p
Output: k synthetic minority rows in the detector space

- 1 fit the quantile codec C on X , set $Z = C.encode(X)$
- 2 $S^+ = \text{rows of } X \text{ with } y = 1$, $Z^+ = C.encode(S^+)$, $S^- = \text{rows of } Z \text{ with } y = 0$
- 3 $k = \text{round}(q \times |S^-|)$
- 4 fit the mixed transform V on S^+ , set $W = V.transform(S^+)$
- 5 fit the conditional VAE on (W, s) and the conditional WGAN-GP on (W, s)
- 6 $n_{\text{pool}} = p \times k$
- 7 allocate scenario codes over the pool by square root reweighting
- 8 draw $\alpha \times n_{\text{pool}}$ rows from the VAE and the rest from the WGAN-GP
- 9 map the pool back through V and then through $C.encode$
- 10 $R = \text{random subset of } S^- \text{ of size } |S^+|$
- 11 for every candidate compute $d_{\min}$ against Z^+ and $d_{\max}$ against R
- 12 keep the candidates with $d_{\min} \leq d_{\max}$ (boundary filter)
- 13 $\delta = 5\text{th percentile of the nearest neighbour distances inside } Z^+$
- 14 drop the candidates with $d_{\min}(1) < \delta$ (novelty filter)
- 15 if fewer than k remain, refill with the candidates that missed by least
- 16 return k rows sampled from those kept

The cost of the procedure is dominated by line 5. Everything after it is one pass of nearest neighbour queries over a pool of a few thousand rows, which takes a fraction of a second.

5. MATERIALS AND METHODS

5.1 Dataset

The CERT r4.2 insider threat corpus (Glasser and Lindauer, 2013) is used. It is a synthetic but carefully constructed record of one thousand employees of a fictional organisation, with logon events, removable media events, file operations, electronic mail and web proxy records, together with a ground truth listing of the insiders and the days on which they acted. The raw event logs are aggregated into one row per employee per day, which is the granularity that has been found to work well for this problem (Le et al., 2020), giving thirty nine behavioural features per user-day. **Table 3** summarises the result and **Figure 4** shows the imbalance, the scenario composition and the column structure that motivated the mixed transform.

Table 3. Summary of the ingested corpus.

Property	Value
Employees	1,000
Employees who are insiders	70
Observation period	2 January 2010 to 17 May 2011
User-days	330,285
Malicious user-days	1,364
Benign user-days	328,921
Imbalance ratio	241.1 to 1
Positive rate	0.41 percent
Behavioural features per user-day	39
Malicious days of scenario 1, 2 and 3	156, 1,188 and 20

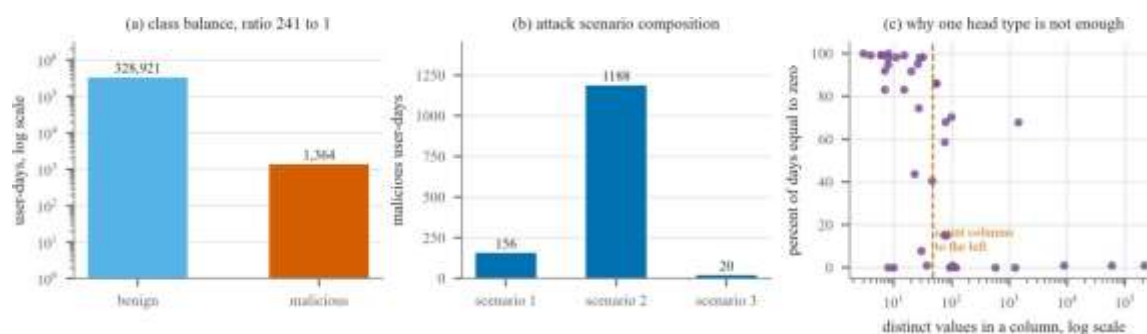


Figure 4. The shape of the problem. Panel (c) plots each column by how many distinct values it takes and how often it is zero, which is why one head type cannot serve all columns.

5.2 Evaluation Protocol

Five fold cross validation is used, and the folds are formed over employees. The insiders and the benign employees are shuffled separately and then interleaved, so that every fold receives insiders. In each fold the employees not under test are split again, one fifth of them forming a validation set that is used only to choose the decision threshold. The threshold that maximises the validation F1 score is then applied unchanged to the test employees. The number of malicious user-days in the test part varied between 167 and 340 across the five folds.

The codec, the generators and the detector are all fitted inside the fold. Synthetic rows are added to the training part only. Validation and test rows are always real. All methods are asked for the same number of synthetic rows, and every synthetic row, whichever method produced it, is passed back to the original scale and re encoded, so that the rounding of counts and the clipping of ranges are applied equally to all of them.

5.3 Baselines

Nine settings are compared. The first is no oversampling at all. The second is cost sensitive class weighting, which is available for the linear model and the forest but not for the neural network used here, so that cell is left empty. Then come random oversampling, SMOTE (Chawla et al., 2002), Borderline SMOTE (Han et al., 2005) and ADASYN (He et al., 2008), all taken from the imbalanced-learn library and all driven to the same target count. The last three are the conditional autoencoder alone, the conditional adversarial network alone and the full HGO. The two single generator settings draw from the same trained pair of networks as HGO, with the mixing share set to one or to zero, so a difference between these three rows is a difference in sampling and not an accident of how the networks happened to converge.

5.4 Detectors and Settings

Three detectors from different model families are used, because an oversampler that helps only one family is not a useful result. Logistic regression stands for the linear family, a Random Forest (Breiman, 2001) for the tree ensemble family, and a multi layer perceptron for the neural family. The settings are listed in **Table 4**. They were fixed before the experiments and were not tuned per oversampler, since the aim is to compare the data and not the classifier.

Table 4. Settings used in all experiments.

Component	Setting
Folds	5, split by employee, insider stratified
Target ratio q	0.01 of the majority count in the main
table Pool factor p	3
Mixing share α	0.5
Neighbourhood size κ	5
Codec	quantile transform to a normal, 1000 knots
Count column limit	48 distinct values
Gate threshold	2 percent zero mass
VAE	encoder 256 and 128, latent 24, β 0.5, 600 epochs

WGAN-GP	generator 256 and 256, critic 256 and 256, 1500
steps WGAN-GP critic ratio	5 critic steps per generator step, λ 10
Gumbel temperature	0.2
Optimiser	Adam, 0.001 for the VAE and 0.0002 for the WGAN-
GP Logistic regression	L2 penalty, $C = 1$, at most 300 iterations
Random Forest	150 trees, minimum leaf size 2
Multi layer perceptron	hidden layers 128 and 64, early
stopping Random seed	20260812

5.5 Metrics

Because the positive rate is under half of one percent, accuracy and the area under the receiver operating characteristic curve are close to useless. The area under the precision recall curve is the primary threshold free measure, as it responds to the change in the positive class that matters here (Davis and Goadrich, 2006; Saito and Rehmsmeier, 2015). Alongside it we report the F1 score, precision, recall, the geometric mean of sensitivity and specificity and the Matthews correlation coefficient at the threshold chosen on the validation employees. The quality of the synthetic rows is measured with four further quantities: the mean Kolmogorov Smirnov statistic between the real and the synthetic value of each column, the relative difference between the two correlation matrices, the cross validated area under the curve of a Random Forest trained to tell real minority rows from synthetic ones, where a value of 0.5 means the two are indistinguishable, and the median distance from a synthetic row to the closest real record expressed as a multiple of the median distance between two real minority rows. A value of that last quantity below one means the synthetic rows sit closer to real records than real records sit to each other, which is the signature of copying.

6. RESULTS AND DISCUSSION

6.1 Detection Performance

Table 5. Detection performance, mean over five employee disjoint folds. The best value in each column is shown in bold.

Oversampler	RF PR- AUC	RF F1	RF recall	RF MCC	LR PR- AUC	LR F1	MLP PR- AUC	MLP F1
None								
Class weighting	0.381	0.369	0.339	0.392	0.192	0.227	0.220	0.246
Random	0.351	0.364	0.279	0.397	0.111	0.141	n/a	n/a
oversampling	0.367	0.393	0.313	0.415	0.192	0.229	0.152	0.208
SMOTE								
Borderline	0.341	0.348	0.325	0.357	0.182	0.219	0.179	0.218
SMOTE	0.330	0.329	0.307	0.350	0.189	0.222	0.170	0.215
ADASYN	0.361	0.381	0.318	0.394	0.180	0.217	0.174	0.207
Conditional VAE	0.390	0.399	0.358	0.421	0.178	0.197	0.180	0.191
Condition	0.360	0.395	0.303	0.426	0.180	0.187	0.175	0.208
al								
WGAN-GP	0.388	0.409	0.324	0.440	0.179	0.206	0.180	0.218
HGO (proposed)								

Table 5 and **Figure 5** give the main result. With the Random Forest detector, HGO obtains the best F1 score, 0.409 against 0.369 without any oversampling, an improvement of 11.1 percent, and the best Matthews correlation, 0.440 against

0.392, an improvement of 12.4 percent. Against the interpolation family the margin is larger. HGO improves on SMOTE by 17.6 percent in F1 and by 13.5 percent in precision recall area, on Borderline SMOTE by 24.4 percent in F1, and on ADASYN by 7.5 percent in F1.

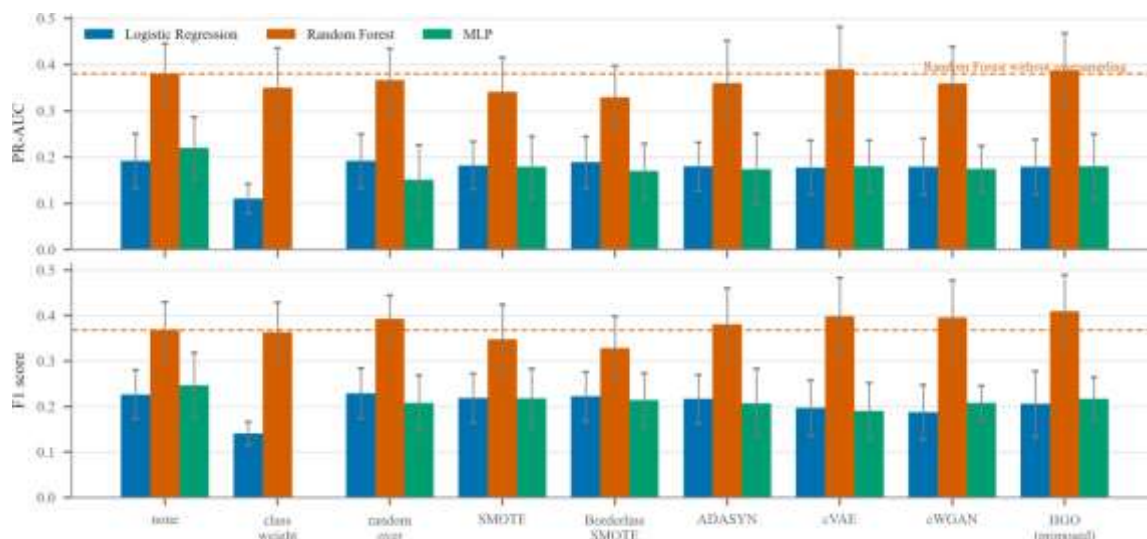


Figure 5. Precision recall area and F1 score for every oversampler and every detector. The bars are means over the five folds and the whiskers are one standard deviation across folds.

The error bars in **Figure 5** are wide, so the ranking has to be read with care. A mean can be carried by one lucky fold. We therefore also counted, for each comparison, the number of folds in which HGO was ahead. Against SMOTE, HGO won the precision recall area in 5 folds out of five, against Borderline SMOTE in 5 and against ADASYN in 5. The advantage over the interpolation family is consistent. Against random oversampling the picture is different: HGO has the better mean, by 4.2 percent in F1, but it wins in only 1 fold out of five, so that difference should not be claimed. Against no oversampling at all, HGO wins the F1 score in 3 folds and the precision recall area in 2, which says that the gain is mostly in where the decision threshold ends up rather than in the ranking itself.

The other two detectors tell a plainer story. For logistic regression and for the multi layer perceptron, no oversampler improved on the plain imbalanced baseline. The linear model scores 0.192 in precision recall area without help and at most 0.192 with it. The neural model scores 0.220 without help and at most 0.180 with it. Among the oversamplers, the generative ones are level with or slightly ahead of the interpolation family on the neural model and slightly behind it on the linear model, but none of them repairs the underlying issue, which is that thirty nine aggregated counts are not enough for these two model families to separate the classes on this corpus. Cost sensitive weighting was actively harmful for the linear model, cutting its precision recall area from 0.192 to 0.111.

6.2 Ranking Behaviour

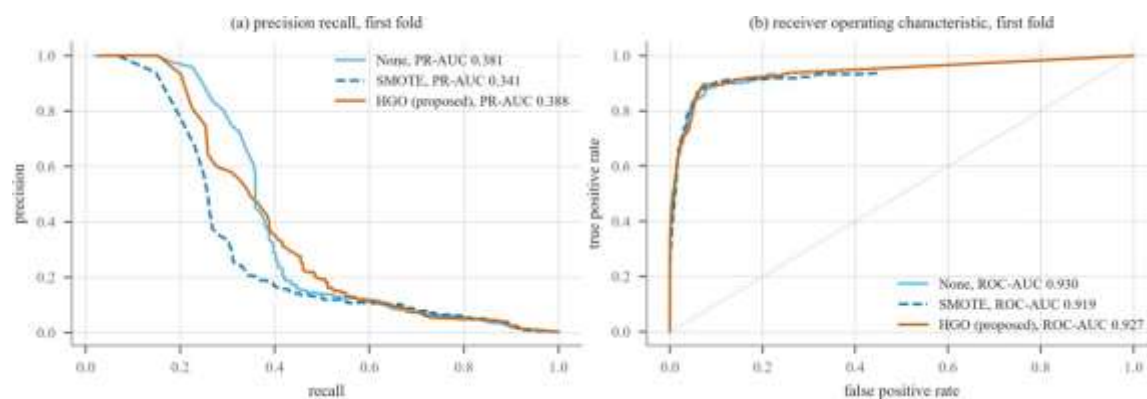


Figure 6. Precision recall and receiver operating characteristic curves for the Random Forest on the first fold.

Figure 6 shows the curves for the first fold. The receiver operating characteristic curves of the three settings are almost on top of each other, which is what the literature on imbalanced evaluation predicts (Davis and Goadrich, 2006; Saito and Rehmsmeier, 2015): with 241 benign days for every malicious one, a large change in the number of false alarms hardly moves the false positive rate. The precision recall curves separate much more clearly, and the curve for HGO stays above the curve for SMOTE over most of the recall range. This is the reason the precision recall area is used as the primary measure throughout.

6.3 How Much Synthetic Data to Add

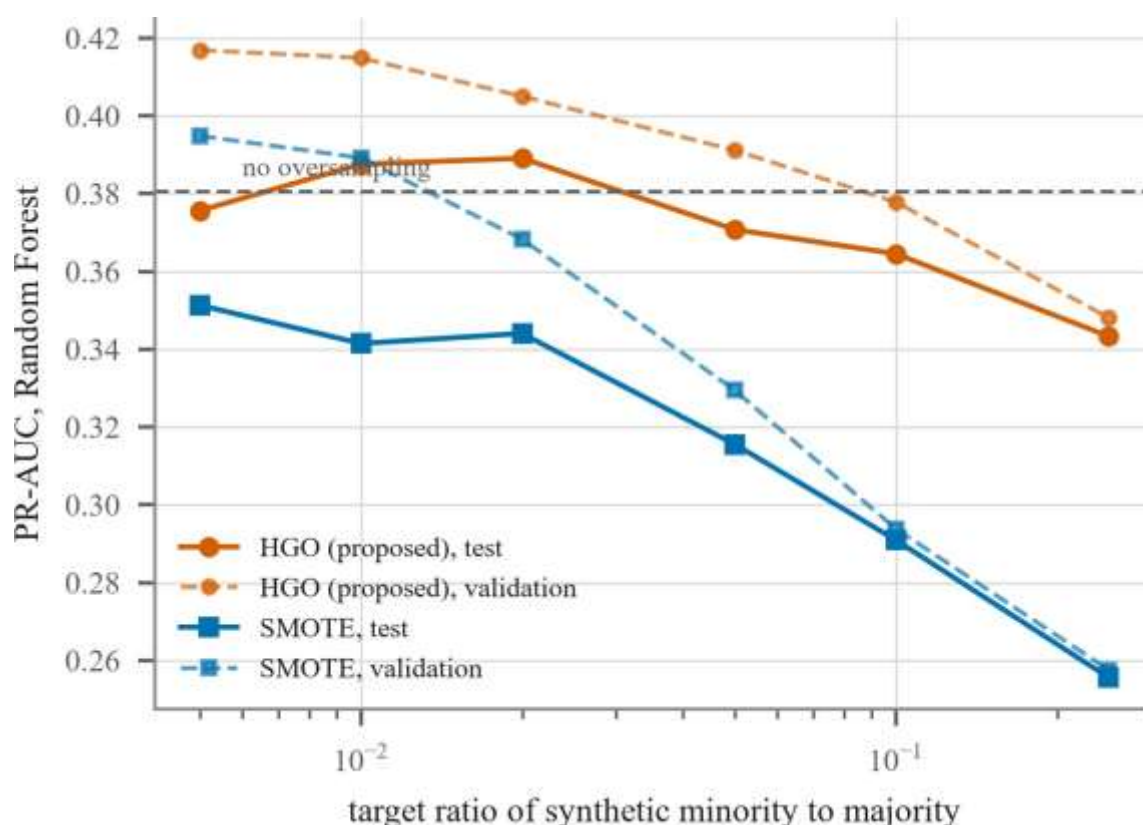


Figure 7. Effect of the target ratio. Solid lines are the held out test employees and dashed lines are the validation employees.

Figure 7 sweeps the target ratio from 0.005 to 0.25 of the majority count, which on this corpus is about one thousand to fifty thousand synthetic rows. Two things are visible. First, more synthetic data is not better. Both methods peak early and then decline, and by $\rho = 0.25$ SMOTE has fallen to 0.256 in precision recall area, well below the 0.381 of the plain baseline, while HGO has fallen only to 0.343. Adding a large amount of synthetic minority data drowns the real signal, and the interpolation family drowns it faster. Second, the validation curve has the same shape as the test curve, which means the ratio can be chosen on validation data without touching the test employees. The value $\rho = 0.01$ used in the main table is the peak of the validation curve. HGO is above SMOTE at every ratio, and in five folds out of five at every ratio from 0.01 upwards.

6.4 Quality of the Synthetic Rows

Table 6. Quality of the synthetic minority rows, mean over five folds. The proposed method is shown in bold.

Method	Mean KS	Correlation difference	Real against synthetic AUC	Distance to closest real record
Random oversampling	0.010	0.071	0.301	0.000
SMOTE	0.026	0.142	0.491	0.315
Borderline SMOTE	0.050	0.190	0.619	0.278

ADASYN	0.036	0.162	0.533	0.325
Conditional VAE	0.047	0.290	0.836	1.264
Conditional WGAN-GP	0.042	0.558	0.936	2.421
HGO (proposed)	0.038	0.373	0.855	1.824

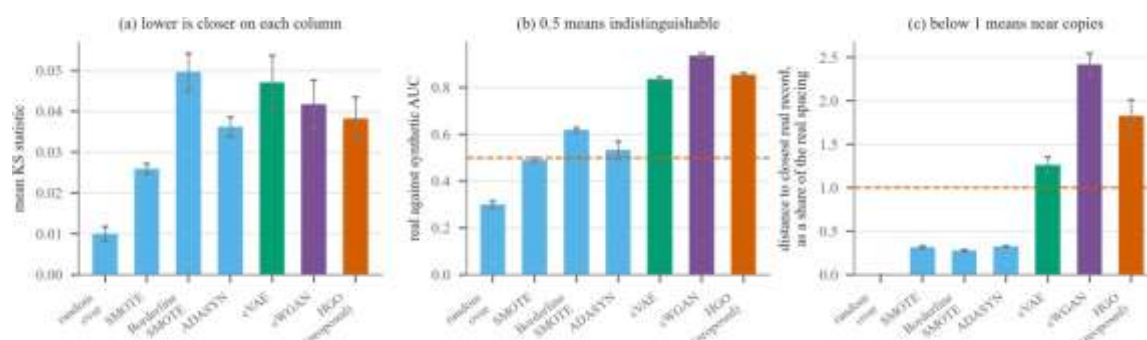


Figure 8. Three views of synthetic data quality. Interpolation scores well on the first two panels for the wrong reason, which the third panel exposes.

Table 6 and **Figure 8** contain the most interesting result in the paper, and it is a warning about how synthetic data is usually judged. On the first two measures the interpolation family looks excellent. SMOTE has a mean Kolmogorov Smirnov statistic of 0.026 and a discriminator that reaches only 0.491, which is chance level, so by the usual reading its rows are indistinguishable from real ones. HGO looks worse on that reading, with a discriminator at 0.855.

The last column explains why. The median distance from a SMOTE row to the closest real record is only 0.315 of the median distance between two real minority rows, and for random oversampling it is 0.000, because those rows are exact duplicates. In other words the interpolation family scores well on indistinguishability because its output is very close to the input. HGO sits at 1.824, above one, so its rows are genuinely new points that are further from any real employee day than real employee days are from each other. The adversarial network alone goes further still, to 2.421, which is novelty bought at the cost of fidelity, and this is visible in its correlation difference of 0.558 against 0.290 for the autoencoder. The hybrid sits between its two parts on both counts, which is the intended behaviour.

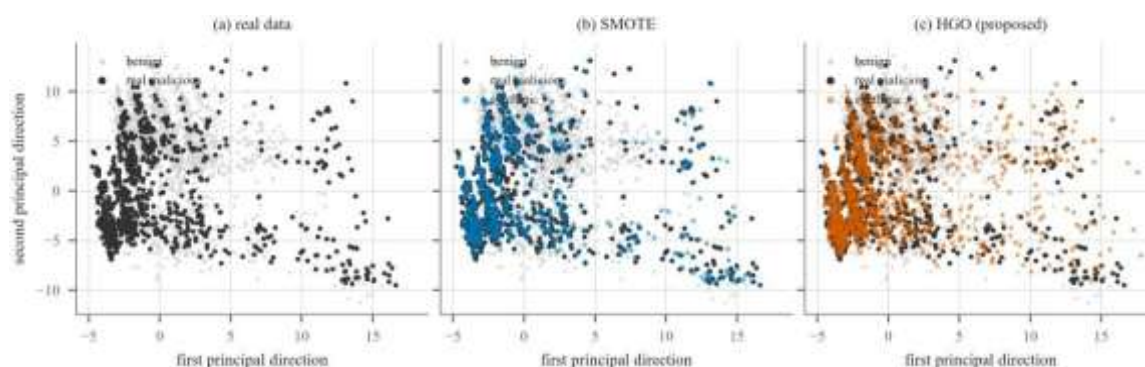


Figure 9. Real and synthetic malicious user-days in the first two principal directions of the first fold. SMOTE rows lie on the real rows, while HGO rows fill the region around them.

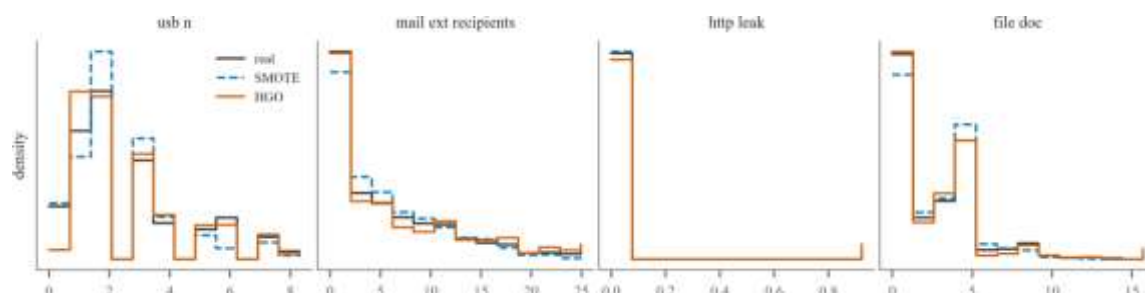


Figure 10. Distribution of four representative columns for real, SMOTE and HGO malicious user-days.

Figure 9 shows the same fact geometrically. In the first two principal directions of the first fold, the SMOTE rows sit almost exactly on the real malicious rows, while the HGO rows spread over the region those rows occupy, including the gaps between them. **Figure 10** confirms that this spreading has not damaged the individual columns: for a count column such as the number of removable media insertions, and for a very rare column such as the number of visits to a data leakage site, the generated distribution follows the real one closely.

6.5 Ablation

Table 7. Ablation study with the Random Forest detector, mean over five folds. The full method is shown in bold.

Setting	PR-AUC	F1	Recall	MCC
no oversampling	0.381	0.369	0.339	0.392
autoencoder only	0.390	0.399	0.358	0.421
adversarial network only	0.360	0.395	0.303	0.426
hybrid without scenario conditioning	0.377	0.382	0.332	0.408
hybrid without the two filters	0.387	0.393	0.325	0.423
full method	0.388	0.409	0.324	0.440

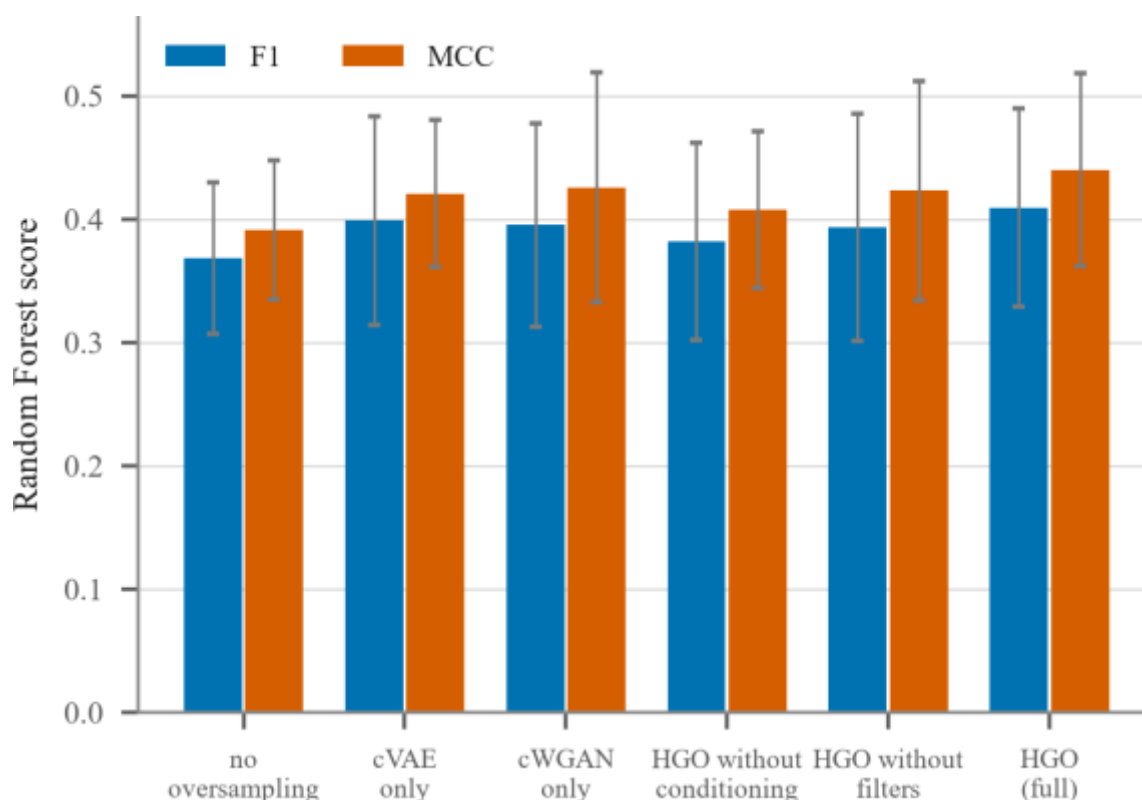


Figure 11. Contribution of each part of the method.

Table 7 and **Figure 11** take the method apart. Removing the two filters costs 3.9 percent of the F1 score and 3.8 percent of the Matthews correlation. Removing the scenario conditioning costs more, 6.7 percent of F1 and 7.3 percent of the Matthews correlation, which places it as the more valuable of the two additions. Each generator on its own is weaker than the pair in F1 score and Matthews correlation: the autoencoder alone gives the highest recall of any setting, 0.358, and a marginally higher precision recall area, 0.390 against 0.388, but converts less of its recall into precision, and the adversarial network alone is behind the hybrid on every measure.

The filters are also worth looking at directly. On the first fold the pool held 6,315 candidates for 2,105 finally kept. The boundary filter removed 1,257 of them, that is about 20 percent, and the novelty filter removed a further 25. The small number rejected for novelty is itself informative: the generators are not copying their training rows, so the novelty filter is a cheap safeguard rather than a major clean up step.

6.6 Cost and Per Scenario Behaviour

Table 8. Cost per fold and behaviour on the rarest scenario.

Item	Value
Time to fit both generators, one fold	118 s
Time to draw and filter the pool	0.13 s
Time to fit SMOTE on the same fold	0.53 s
Whole study, five folds, all settings	58 min
Recall on scenario 1, no oversampling against HGO	0.471 and 0.471
Recall on scenario 2, no oversampling against HGO	0.327 and 0.313
Recall on scenario 3, no oversampling against HGO	0.533 and 0.383

Table 8 collects the cost of the method. Fitting the two networks takes about 118 seconds per fold on an ordinary desktop processor with no graphics card, against a fraction of a second for SMOTE. Drawing and filtering the pool afterwards is almost free. Whether the trade is worth making depends on how often the generator has to be refitted. In a deployment where the model is retrained weekly, two minutes is nothing. In a study that sweeps many configurations, it adds up.

We had expected the scenario conditioning to show up as a gain on the third scenario, which is the rarest. It did not. The whole corpus contains only 20 malicious user-days of that scenario, which is four per fold on average, and the recall on it moved from 0.533 without oversampling to 0.383 with HGO. With four positive days per fold, neither number carries any weight, and we report the comparison only for completeness. The value of the conditioning shows instead in the ablation, where removing it costs 6.7 percent of the F1 score across all scenarios. The most likely explanation is that conditioning gives the generators a cleaner target, since the three scenarios are genuinely different kinds of behaviour, and mixing them into one distribution makes that distribution harder to model.

6.7 Discussion and Threats to Validity

Three limits should be stated plainly. The first is statistical. Five folds over seventy insiders is a small sample, the fold to fold spread is wide, and several of the differences reported above are smaller than one standard deviation. This is why the number of folds won is given alongside every mean, and why the comparison against random oversampling is described as unproven rather than as a result.

The second limit is the corpus. CERT r4.2 is generated data. The benign behaviour comes from a simulator and the malicious behaviour was inserted deliberately, so the malicious rows may be more internally consistent than real insider activity would be, which would make them easier to model and would flatter any generative method. The findings should be read as evidence about this benchmark and as a method that is worth trying elsewhere, not as a settled result about real organisations.

The third limit is the feature set. This study deliberately used the thirty nine raw behavioural counts and nothing else, so that the effect of the oversampler would not be mixed up with the effect of feature engineering. Richer representations, in particular per employee baselines and rolling windows, are known to lift absolute detection performance considerably. Whether generative oversampling still helps once those are present is an open question, and it is the natural next experiment.

One practical recommendation follows from the results. If the detector is a tree ensemble and the operating point matters, which is the usual situation in a security operations centre, then generative oversampling at a small ratio is worth the two minutes it costs. If the goal is only to rank alerts and the detector already handles imbalance internally, the plain baseline is

hard to beat and the simplest option is to leave the data alone.

7. CONCLUSION AND FUTURE WORK

This paper studied whether deep generative models are a better answer to extreme class imbalance in insider threat detection than the interpolation based oversamplers that are normally used. The proposed method, Hybrid Generative Oversampling, fits a conditional variational autoencoder and a conditional Wasserstein generative adversarial network on the malicious rows of each training fold, gives every column of the user-day the head that matches its type, draws a candidate pool from both networks and then removes the candidates that fall inside the benign region or that copy a real record.

On the CERT r4.2 corpus, with an imbalance of 241 to 1 and an employee disjoint five fold protocol, the method raised the Random Forest F1 score from 0.369 to 0.409 and the Matthews correlation from 0.392 to 0.440, and it improved on SMOTE in precision recall area in five folds out of five and at every oversampling ratio tested. Its synthetic rows sit 1.824 times the real spacing away from the nearest real record, against 0.315 for SMOTE, so the gain does not come from copying. Equally clearly, no oversampler helped the linear and neural detectors on this data, and the advantage over plain random oversampling was not consistent across folds.

Three directions follow. The first is to combine the generated data with richer temporal features, since the sequence in which a person's days change is more informative than any single day, and an ensemble of an isolation forest (Liu et al., 2008) with a long short term memory network over those sequences is the subject of our current work. The second is to test the method on a corpus of real logs, where the malicious behaviour is messier. The third is to make the novelty filter a formal privacy guarantee rather than a distance threshold, which would allow a generated corpus to be shared between organisations, something the field badly needs given how few insider threat datasets exist.

ACKNOWLEDGEMENTS

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

AUTHOR CONTRIBUTIONS

Mahesh Kumar Anem: Conceptualization, Methodology, Software, Formal Analysis, Visualization, Writing (Original Draft).

Abdul Ahad: Conceptualization, Supervision, Validation, Writing (Review & Editing).

All authors read and approved the final manuscript.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

ETHICS STATEMENT

This study used only the publicly available CERT r4.2 insider threat dataset, which is synthetically generated and contains no records of real persons. No human participants or animals were involved, and ethical approval and informed consent were therefore not required. No generative artificial intelligence tool was used to produce the scientific content, the analysis or the results reported here.

DATA AVAILABILITY STATEMENT

The CERT r4.2 insider threat dataset is publicly available from the CERT Division of the Software Engineering Institute, Carnegie Mellon University (Lindauer, 2020). The analysis code and result files are available from the corresponding author on reasonable request.

REFERENCES

- [1] Arjovsky, M., Chintala, S., Bottou, L., 2017. *Wasserstein generative adversarial networks*, In: Proceedings of the 34th International Conference on Machine Learning (ICML). PMLR 70, Sydney, Australia, pp. 214-223.
- [2] Breiman, L., 2001. *Random forests*. Machine Learning 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324>

- [3] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. *SMOTE: Synthetic minority over-sampling technique*. Journal of Artificial Intelligence Research 16, 321-357. <https://doi.org/10.1613/jair.953>
- [4] Davis, J., Goadrich, M., 2006. *The relationship between precision-recall and ROC curves*, In: Proceedings of the 23rd International Conference on Machine Learning (ICML). ACM, Pittsburgh, PA, USA, pp. 233-240. <https://doi.org/10.1145/1143844.1143874>
- [5] Douzas, G., Bacao, F., 2018. *Effective data generation for imbalanced learning using conditional generative adversarial networks*. Expert Systems with Applications 91, 464-471. <https://doi.org/10.1016/j.eswa.2017.09.030>
- [6] Engelmann, J., Lessmann, S., 2021. *Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning*. Expert Systems with Applications 174, 114582. <https://doi.org/10.1016/j.eswa.2021.114582>
- [7] Fernández, A., García, S., Herrera, F., Chawla, N.V., 2018. *SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary*. Journal of Artificial Intelligence Research 61, 863-905. <https://doi.org/10.1613/jair.111192>
- [8] Glasser, J., Lindauer, B., 2013. *Bridging the gap: A pragmatic approach to generating insider threat data*, In: Proceedings of the IEEE Security and Privacy Workshops (SPW). IEEE, San Francisco, CA, USA, pp. 98-104. <https://doi.org/10.1109/SPW.2013.37>
- [9] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. *Generative adversarial nets*, In: Advances in Neural Information Processing Systems 27. Montreal, Canada, pp. 2672-2680.
- [10] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A., 2017. *Improved training of Wasserstein GANs*, In: Advances in Neural Information Processing Systems 30. Long Beach, CA, USA, pp. 5767-5777.
- [11] Han, H., Wang, W.-Y., Mao, B.-H., 2005. *Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning*, In: Advances in Intelligent Computing, Lecture Notes in Computer Science 3644. Springer, Berlin, Germany, pp. 878-887. https://doi.org/10.1007/11538059_91
- [12] He, H., Bai, Y., Garcia, E.A., Li, S., 2008. *ADASYN: Adaptive synthetic sampling approach for imbalanced learning*, In: Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN). IEEE, Hong Kong, pp. 1322-1328. <https://doi.org/10.1109/IJCNN.2008.4633969>
- [13] He, H., Garcia, E.A., 2009. *Learning from imbalanced data*. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263-1284. <https://doi.org/10.1109/TKDE.2008.239>
- [14] Homoliak, I., Toffalini, F., Guarnizo, J., Elovici, Y., Ochoa, M., 2019. *Insight into insiders and IT: A survey of insider threat taxonomies, analysis, modeling, and countermeasures*. ACM Computing Surveys 52(2), 30. <https://doi.org/10.1145/3303771>
- [15] Jang, E., Gu, S., Poole, B., 2017. *Categorical reparameterization with Gumbel-softmax*, In: Proceedings of the 5th International Conference on Learning Representations (ICLR). Toulon, France. <https://doi.org/10.48550/arXiv.1611.01144>
- [16] Kingma, D.P., Welling, M., 2014. *Auto-encoding variational Bayes*, In: Proceedings of the 2nd International Conference on Learning Representations (ICLR). Banff, Canada. <https://doi.org/10.48550/arXiv.1312.6114>
- [17] Krawczyk, B., 2016. *Learning from imbalanced data: Open challenges and future directions*. Progress in Artificial Intelligence 5(4), 221-232. <https://doi.org/10.1007/s13748-016-0094-0>
- [18] Le, D.C., Zincir-Heywood, N., Heywood, M.I., 2020. *Analyzing data granularity levels for insider threat detection using machine learning*. IEEE Transactions on Network and Service Management 17(1), 30-44. <https://doi.org/10.1109/TNSM.2020.2967721>
- [19] Lindauer, B., 2020. *Insider Threat Test Dataset [Dataset]*. Carnegie Mellon University. <https://doi.org/10.1184/R1/12841247.v1>
- [20] Liu, F.T., Ting, K.M., Zhou, Z.-H., 2008. *Isolation forest*, In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM). IEEE, Pisa, Italy, pp. 413-422. <https://doi.org/10.1109/ICDM.2008.17>
- [21] Mirza, M., Osindero, S., 2014. *Conditional generative adversarial nets*. arXiv preprint arXiv:1411.1784. <https://doi.org/10.48550/arXiv.1411.1784>
- [22] Saito, T., Rehmsmeier, M., 2015. *The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets*. PLOS ONE 10(3), e0118432. <https://doi.org/10.1371/journal.pone.0118432>

- [23] Sharma, A., Singh, P.K., Chandra, R., 2022. *SMOTified-GAN for class imbalanced pattern classification problems*. IEEE Access 10, 30655-30665. <https://doi.org/10.1109/ACCESS.2022.3158977>
- [24] Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., Robinson, S., 2017. *Deep learning for unsupervised insider threat detection in structured cybersecurity data streams*, In: Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security. San Francisco, CA, USA, pp. 224-231.
- [25] Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K., 2019. *Modeling tabular data using conditional GAN*, In: Advances in Neural Information Processing Systems 32. Vancouver, Canada, pp. 7335-7345.
- [26] Yuan, S., Wu, X., 2021. *Deep learning for insider threat detection: Review, challenges and opportunities*. Computers & Security 104, 102221. <https://doi.org/10.1016/j.cose.2021.102221>