Journal of Software Engineering and Applications, 2023, 16, 586-604
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Generative Adversarial Network Based Approach towards Synthetically Generating Insider Threat Scenarios

Mayesh Mohapatra[1], Anshumaan Phukan[2], Vijay K. Madisetti[3]

[1]Indian Institute of Science, Bengaluru, India
[2]Bennett University, Greater Noida, India
[3]School of Cybersecurity and Privacy, Georgia Institute of Technology, Atlanta, USA
Email: mayeshm@iisc.ac.in, e19cse062@bennett.edu.in, vkm@gatech.edu

How to cite this paper: Mohapatra, M., Phukan, A. and Madisetti, V.K. (2023) Generative Adversarial Network Based Approach towards Synthetically Generating Insider Threat Scenarios. *Journal of Software Engineering and Applications*, **16**, 586-604.
https://doi.org/10.4236/jsea.2023.1611030

Received: October 5, 2023
Accepted: November 25, 2023
Published: November 28, 2023

Abstract

This research paper explores the use of Generative Adversarial Networks (GANs) to synthetically generate insider threat scenarios. Insider threats pose significant risks to IT infrastructures, requiring effective detection and mitigation strategies. By training GAN models on historical insider threat data, synthetic scenarios resembling real-world incidents can be generated, including various tactics and procedures employed by insiders. The paper discusses the benefits, challenges, and ethical considerations associated with using GAN-generated data. The findings highlight the potential of GANs in enhancing insider threat detection and response capabilities, empowering organizations to fortify their defenses and proactively mitigate risks posed by internal actors.

Keywords

GANs, CERT, Insider-Threat, Cybersecurity

1. Introduction

Insider threats are a major concern for both small-scale and large-scale IT infrastructures. As organizations increasingly rely on technological systems to store, process, and transmit sensitive information, the potential for internal actors to exploit their authorized access poses a significant risk to data security and operational integrity. These insider threats encompass a wide range of malicious activities, including theft of intellectual property, unauthorized disclosure of sensitive data, sabotage of critical systems, and fraud. The impact of such threats can be severe, leading to financial losses, reputational damage, legal consequences, and

disruption of business operations. Consequently, understanding the nature and dynamics of insider threats has become a critical area of research and practical implementation for organizations aiming to protect their valuable assets and maintain a robust security posture.

In addition to the malicious intentions of individuals in an organization, it is crucial to explore the methods employed by insiders to exploit their authorized access. Insider threats often leverage their knowledge of internal systems, processes, and security protocols to bypass traditional safeguards and carry out their malicious activities. These methods may involve circumventing access controls, utilizing privileged accounts, abusing administrative privileges, or employing sophisticated techniques such as social engineering. By comprehending the tactics used by insiders, organizations can implement robust controls and monitoring mechanisms to detect and respond to potential threats in a timely manner.

One valuable resource that aids in understanding and addressing insider threats is the CERT Insider Threat Dataset [1] developed by the Computer Emergency Response Team at the Software Engineering Institute at Carnegie Mellon University. The dataset provides a comprehensive collection of real-world insider threat incidents. It draws from anonymized and aggregated information derived from incident reports, law enforcement case files, and internal investigations. This rich dataset allows researchers to analyze the dataset and extract valuable insights regarding common attack vectors, insider profiles, indicators of compromise, and the impact of insider threats on organizations.

The CERT Insider Threat Dataset has helped researchers to delve into the motivations that drive individuals to engage in insider threats, the pre-incident behaviors, and the methods employed to carry out malicious activities. This information aids in understanding the psychological, behavioral, and situational factors that contribute to these insider threat incidents.

The dataset also serves as a crucial resource for the development and validation of insider threat detection and mitigation techniques. Researchers can use the dataset to test and refine their algorithms, compare different detection approaches, and evaluate the effectiveness of insider threat prevention strategies. This contributes to the advancement of insider threat research and the development of robust models and tools for addressing insider threats.

The CERT Dataset, although a valuable resource for studying insider threats, is not exempt from the challenges associated with class imbalance. Class imbalance refers to the unequal distribution of samples among different classes or categories within a dataset. In the case of the CERT Insider Threat Dataset, class imbalance arises from the disparity between the number of instances representing insider threat incidents and the number of instances representing non-threat or normal behavior.

The class imbalance problem in the CERT Insider Threat Dataset can have significant implications for developing effective insider threat detection models. Traditional machine learning algorithms tend to be biased toward the majority

class, which in this case would be the non-threat instances. As a result, these algorithms may exhibit a reduced sensitivity and accuracy in detecting and predicting insider threats, as they prioritize optimizing their performance on the majority class (Table 1).

The consequences of class imbalances in the CERT Insider Threat Dataset are twofold. First, the limited representation of insider threat instances hinders the model's ability to learn and generalize from these rare but critical instances. This means that the model may struggle to accurately identify and distinguish insider threats from normal behavior, leading to a higher rate of false negatives (missed threats) and lower overall detection performance.

Secondly, when developing insider threat detection models, it is crucial to acknowledge the dynamic nature of insider threats and the potential for attackers to employ new scenarios that are not initially present in the dataset used for training or evaluation. Insider threat incidents can vary in their tactics, techniques, and procedures (TTPs), and threat actors constantly evolve their strategies to bypass detection mechanisms.

The CERT Insider Threat Dataset, although comprehensive, may not encompass all possible insider threat scenarios. Therefore, it is important to consider the limitations of the dataset and the need for adaptability and continuous learning in insider threat detection models.

In recent years, the application of Generative Adversarial Networks (GANs) has emerged as a promising approach to address the challenges of insider threats. GANs are a class of machine learning models that consist of a generator network and a discriminator network, which are trained in an adversarial manner. The generator learns to synthesize realistic data samples, while the discriminator network aims to distinguish between real and synthetic data.

The use of GANs in the context of insider threats involves synthetically generating insider threat scenarios. By training GAN models on the CERT Insider Threat Dataset and other relevant data sources, researchers can generate realistic insider threat instances that mimic the characteristics and patterns observed in real-world incidents. This synthetic data generation enables researchers to explore various insider threat scenarios, test the effectiveness of detection and mitigation techniques, and develop robust models to combat insider threats.

One advantage of using GANs for generating insider threat scenarios is the ability to create a diverse set of data samples. GANs can capture the complex and multidimensional nature of insider threats by learning the underlying distribution of the data. This allows researchers to generate a wide range of synthetic

Table 1. Number of scenarios in each version of the CERT insider threat dataset.

Version	Normal	Sc.1, Sc.2, Sc.3, Sc.4, Sc.5	Total (Sum(Sc.1 to Sc.5))
CERT r4.2	307,057	85, 861, 20, 0, 0	966
CERT r5.2	647,441	85, 863, 20, 339, 0	1307
CERT r6.2	1,304,406	3, 20, 2, 1, 1	27

instances, encompassing different insider profiles, attack vectors, and behavioral patterns. The diversity of the generated data facilitates a comprehensive analysis of insider threats, enabling the identification of subtle patterns and the exploration of various threat scenarios that may not be well-represented in the original dataset. Furthermore, GANs offer the flexibility to manipulate and control specific aspects of the generated insider threat scenarios. Researchers can introduce variations in the synthetic data to simulate different attack strategies, insider behaviors, or system conditions. This capability allows for the exploration of "what-if" scenarios, assessing the impact of different factors on the success or failure of insider threats. By systematically modifying the generated data, researchers can gain insights into the effectiveness of countermeasures and the resilience of security systems against various insider threat scenarios.

The use of GANs in generating synthetic insider threat scenarios also addresses the challenge of limited data availability. Insider threat incidents are relatively rare compared to other cybersecurity events, making it challenging to obtain a large and diverse dataset for analysis. By leveraging GANs to generate synthetic data, researchers can overcome this limitation and create a more extensive dataset that reflects the complexities and nuances of insider threats. This augmented dataset allows for more robust training, evaluation, and validation of insider threat detection and mitigation techniques.

However, it is important to note that the use of GANs in the context of insider threats is still an evolving area of research. While GANs offer promising capabilities for generating synthetic data, there are challenges and considerations that need to be addressed. These include ensuring the realism and fidelity of the generated data, maintaining the privacy and confidentiality of sensitive information, and addressing potential biases or limitations in the training process.

2. Survey of Recent Approaches

Researchers have proposed a method that utilizes data augmentation to generate synthetic data for training deep learning models in insider threat detection. This method employs a type of deep learning model known as a Generative Adversarial Network (GAN) [2], which comprises two neural networks: a generator and a discriminator. The generator is responsible for generating synthetic data, while the discriminator distinguishes between real and synthetic data. These two networks are trained simultaneously, with the generator aiming to produce synthetic data that is indistinguishable from real data, and the discriminator striving to accurately classify whether the data is real or synthetic.

The researchers utilized a variant of GAN known as the Wasserstein GAN with Gradient Penalty (WCGAN-GP) [3] to produce synthetic data for the purpose of insider threat detection. WCGAN-GP is a type of GAN that incorporates a gradient penalty term in the loss function in order to enhance training stability and mitigate mode collapse, a frequently encountered issue in GAN training. The researchers employed WCGAN-GP to generate synthetic data that accurately emulates genuine user behavior, encompassing activities such as system

login, file access, and network activity.

To assess the effectiveness of their method, the researchers conducted experiments using real-world insider threat datasets. They compared the performance of an insider threat detection model trained solely on real data to a model trained on a combination of real and synthetic data generated using WCGAN-GP. They discovered that the model trained on the combined dataset outperformed the model trained solely on real data, even when only a small amount of labeled data was available for training.

The most notable research being conducted in this field is by Mack Preston [4]. He proposed a methodology involving three major steps: User Behavior Representation, Data Augmentation, and Classification.

Preston's method, which was proposed and put into practice, was used to preprocess the CERT dataset. The proposed technique involved parsing events from log data and sorting them by user and time. For each day and each user, different combinations of statistical and count-based attributes are calculated for feature extraction. The extracted daily features for each user are then processed through a sliding window, which computes the percentile of the current day relative to the trailing window for the same user. A basic normalization of the feature values was implemented to ensure stability and performance of GANs.

Preston implemented various under and up-sampling strategies to solve massive class imbalance issues related to the CERT dataset. Most notably, SMOTE, GAN, CGAN, and WCGAN were explored for oversampling.

The provided research discusses various techniques related to addressing class imbalance and improving the training of Generative Adversarial Networks (GANs):

1) SMOTE (Synthetic Minority Oversampling Technique) [5]: SMOTE is a technique proposed by Chawla *et al.* to address class imbalance. It generates synthetic samples by interpolating between randomly selected minority samples and their nearest neighbors. The implementation in this study utilizes the Python scikit-learn library to balance the class distribution in the dataset.

2) Generative Adversarial Networks (GANs): GANs are a class of neural networks that consist of a generator and a discriminator. They are trained in a min-max game, where the generator aims to generate realistic synthetic samples, and the discriminator tries to distinguish between real and generated samples. The loss function is formulated as a min-max optimization problem that minimizes the Kullback-Leibler divergence between the distributions of real and generated samples.

3) Conditional GANs (CGANs): CGANs extend the GAN architecture by introducing auxiliary data inputs, such as class labels, to both the discriminator and generator. This stabilizes the training process and allows the generation of samples from specific classes, which is beneficial for augmenting imbalanced datasets.

4) Wasserstein GANs (WGANs): WGANs replace the KL-divergence with the Wasserstein distance as the optimization objective. The discriminator in WGANs, called a critic, assigns a score to the realness of samples. This formulation provides more stable training and avoids the vanishing gradient problem.

Gradient penalty is proposed as an alternative to gradient clipping to enforce the Lipschitz constraint, improving stability further (Algorithm 1, Figure 1, Figure 2).

Algorithm 1 WGAN with gradient penalty. We use default values of $\lambda = 10$, $n_{\text{critic}} = 5$, $\alpha = 0.0001$, $\beta_1 = 0$, $\beta_2 = 0.9$.

Require: The gradient penalty coefficient λ, the number of critic iterations per generator iteration n_{critic}, the batch size m, Adam hyperparameters α, β_1, β_2.
Require: initial critic parameters w_0, initial generator parameters θ_0.
while θ has not converged **do**
  for $t = 1, \ldots, n_{\text{critic}}$ **do**
    for $i = 1, \ldots, m$ **do**
      Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$, latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\epsilon \sim U[0, 1]$.
      $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z})$
      $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\tilde{\boldsymbol{x}}$
      $L^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
    end for
    $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2)$
  end for
  Sample a batch of latent variables $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m} \sim p(\boldsymbol{z})$.
  $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -D_w(G_\theta(\boldsymbol{z})), \theta, \alpha, \beta_1, \beta_2)$
end while

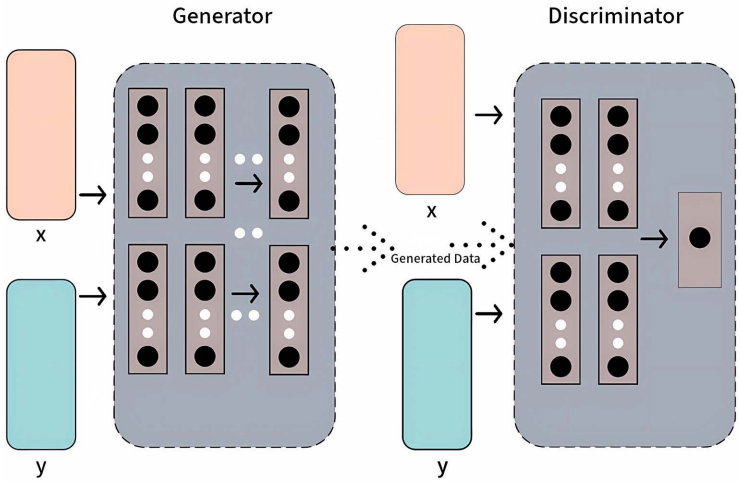

Figure 1. Architecture of Wasserstein GANbib12.

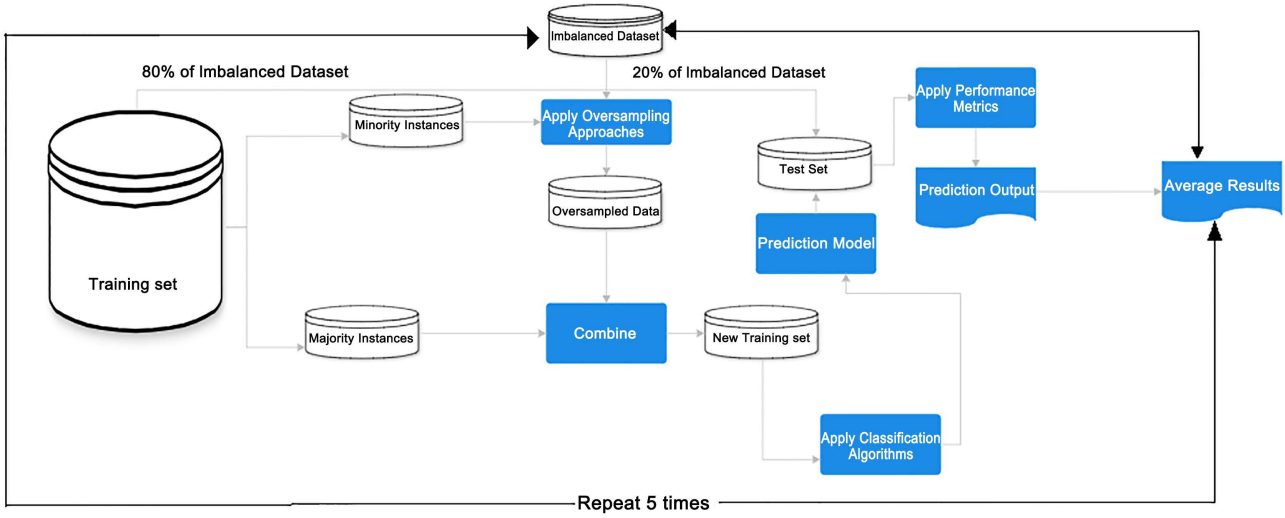

Figure 2. Training setup for Wasserstein GANs with Gradient Penalty.

Four popular classifiers were chosen for testing the data augmentation strategies: Logistic Regression, Random Forest (RF), Support Vector Machines (SVM), and Extreme Gradient Boosting (XGBoost). These classifiers were selected based on their diversity of approach and widespread usage. The XGboost package for Python and the scikit-learn implementations of RF, SVM, and Logistic Regression were used. It discusses the performance of each classifier in an initial testing phase. The classifiers were trained on the augmented R4.2 training datasets, and their performance was evaluated using various evaluation metrics. This phase aimed to assess the effectiveness of the classifiers on the augmented data.

Furthermore, the researchers conducted an ablation study to evaluate the impact of different components of their method on the performance of the insider threat detection model. They found that using WCGAN-GP for data augmentation significantly improved the model's performance compared to other data augmentation methods, such as oversampling and the synthetic minority oversampling technique (SMOTE).

In conclusion, the use of WCGAN-GP for data augmentation in insider threat detection has shown promising results. It has the potential to improve the accuracy of insider threat detection models, especially when labeled data is scarce. However, further research is needed to explore the generalizability and scalability of this method for use in large-scale insider threat detection systems.

3. Proposed Approaches to Improving the Recent Approach

One of the key machine learning algorithms used in this methodology is Support Vector Machines (SVMs) [6]. This model is a powerful and widely used machine learning algorithm for classification and regression tasks. One of the key components of SVM is the kernel function, which allows SVMs to operate effectively in high-dimensional feature spaces. This research carried out additional experiments on different kernel types employed in SVM, their characteristics, and their impact on the performance of SVM algorithms. The aim is to assist researchers and practitioners in understanding the strengths and limitations of various kernel functions [7] and guide them in selecting the most suitable kernel type for specific applications.

The following kernel functions were implemented for the solution:

1) Linear Kernel

The linear kernel computes the dot product between two feature vectors in the original input space. It is defined as:

$$K(x, y) = x \cdot y$$

where x and y represent the feature vectors. The dot product captures the similarity or dissimilarity between the vectors based on their alignment in the feature space. When the dot product is large, it indicates a high degree of similarity between the vectors, while a small dot product suggests dissimilarity.

The linear kernel is suitable for linearly separable data, where the classes can be separated by a hyperplane. It defines a linear decision boundary in the input

space that separates the data points belonging to different classes. The advantage of the linear kernel is its simplicity and efficiency. Since it operates in the original feature space, the computational complexity remains low, making it suitable for large-scale datasets.

2) Polynomial Kernel

The polynomial kernel is defined as:

$$K(x,y)=\left(\gamma*\left(x\cdot y\right)+r\right)^{d}$$

where x and y represent the feature vectors, γ is a scaling parameter, r is an optional coefficient, and d is the degree of the polynomial. The dot product $\left(x\cdot y\right)$ captures the similarity between the vectors, and the polynomial function raises this value to the power of d.

The polynomial kernel allows SVM to capture non-linear relationships between the data points by mapping them to a higher-dimensional space. The degree parameter d determines the complexity of the polynomial function and the flexibility of the decision boundary. The polynomial kernel is useful for datasets that exhibit polynomial patterns or curvature in the decision boundary. It enables SVM to find non-linear decision boundaries that can separate classes with complex relationships. Compared to the linear kernel, the polynomial kernel allows SVM to handle more challenging classification tasks where the data is not linearly separable.

3) RBF

The Radial Basis Function (RBF) kernel is defined as:

$$K(x,y)=\exp\left(-\gamma*\left\|x-y\right\|^{2}\right)$$

where x and y represent the feature vectors, γ is a scaling parameter, $x-y$ is the Euclidean distance between the vectors, and exp denotes the exponential function. The RBF kernel computes the similarity between the feature vectors based on their distance in the input space.

The RBF kernel assigns higher similarity values to feature vectors that are closer to each other, effectively capturing local patterns and relationships in the data. It allows SVM to construct non-linear decision boundaries that can separate classes with complex and overlapping distributions.

One advantage of the RBF kernel is its ability to handle data with irregular shapes and non-linear relationships. It is capable of capturing intricate decision boundaries, making it suitable for a wide range of classification problems.

4) Sigmoid Kernel

The Sigmoid kernel is defined as:

$$K(x,y)=\tanh\left(\gamma*\left(x\cdot y\right)+r\right)$$

where x and y represent the feature vectors, γ is a scaling parameter, $\left(x\cdot y\right)$ denotes the dot product between the vectors, r is an optional coefficient, and tanh is the hyperbolic tangent function. The Sigmoid kernel computes the similarity between feature vectors based on the hyperbolic tangent of a linear com-

bination of their dot product.

The Sigmoid kernel is useful for handling non-linearly separable data, especially when the classes exhibit a sigmoidal or S-shaped pattern. It allows SVM to find decision boundaries that are curved or sigmoidal in shape, capturing complex relationships between data points.

However, it is important to note that the Sigmoid kernel can be sensitive to parameter tuning. The scaling parameter γ determines the influence of the dot product, while the coefficient r controls the bias or offset of the decision boundary.

By thoroughly examining the characteristics, advantages, and limitations of linear, polynomial, RBF, and sigmoid kernels in SVM, this research aims to provide a comprehensive understanding of the various kernel types and their impact on SVM performance. It enables researchers and practitioners to make informed decisions regarding kernel selection, leading to improved accuracy and efficiency in their machine learning tasks.

Random Forest:

Random Forest is a popular ensemble learning method implemented. It combines multiple decision trees to enhance classification performance and is widely used in various domains, including insider threat detection, due to its robustness and ability to handle complex datasets. In this study, we delved deep into the key parameters of Random Forest to increase model performance. The parameters we experimented with are as follows.

Number of Trees:

The number of trees in a Random Forest ensemble is a crucial parameter that determines the trade-off between model performance and computational efficiency. Increasing the number of trees generally improves the model's predictive accuracy, as it reduces the impact of individual noisy or biased trees.

Tree Depth (Maximum Depth):

The maximum depth parameter controls the depth of individual decision trees within the Random Forest. A deeper tree can capture more intricate relationships in the data but is also more prone to overfitting.

Feature Subsampling:

Random Forest models randomly select a subset of features for each individual tree. The feature subsampling parameter determines the proportion of features to be considered at each split. By randomly selecting subsets of features, Random Forest can reduce the correlation between trees and improve the overall diversity of the ensemble.

Sample Subsampling:

Random Forest also performs sampling of the training data for each tree. This technique, known as bagging or bootstrap aggregating, involves randomly selecting a subset of training instances with replacement. The sample subsampling parameter controls the proportion of training instances used for each tree.

Splitting Criteria:

The splitting criterion determines how the decision trees in a Random Forest split the data at each node. The two commonly used criteria are Gini impurity and entropy (information gain). Gini impurity measures the probability of misclassifying a randomly chosen instance, while entropy considers the information gain in terms of class distribution.

Several generative models were implemented in our for AI-based insider threat detection system, including Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs), and Variational Autoencoders (VAEs).

Principal Component Analysis (PCA)

Another major data pre-processing technique used in this study is called Principal Component Analysis (PCA). PCA [8] is a fundamental technique in the field of machine learning and data analysis. It offers a powerful approach to handle high-dimensional datasets by reducing their dimensionality while preserving the most significant patterns and structures within the data.

At its core, PCA aims to transform a dataset consisting of several correlated variables into a new set of uncorrelated variables called principal components. These components are linear combinations of the original variables and are determined based on the directions in the data that capture the maximum amount of variation. By arranging the components in descending order of the variance they explain, PCA ensures that the most important patterns in the data are captured by the first few components. Consequently, one can effectively reduce the dimensionality of the dataset by selecting a subset of these components while retaining most of the information.

The essence of PCA lies in computing the covariance matrix or performing the singular value decomposition of the data matrix. The covariance matrix quantifies the relationships between the variables in the dataset. Through decomposing the covariance matrix, PCA determines the eigenvalues and eigenvectors that define the principal components. The eigenvalues represent the amount of variance explained by each component, while the eigenvectors specify the direction of these components. The eigenvector corresponding to the largest eigenvalue represents the first principal component, which captures the most variation in the data. The subsequent eigenvectors correspond to subsequent principal components, each explaining a decreasing amount of variance.

PCA offers numerous advantages and applications. One of its primary benefits is dimensionality reduction, particularly for datasets with a high number of variables. By selecting a subset of the principal components that capture the most significant variation, one can simplify the dataset while preserving its essential features. This dimensionality reduction leads to several advantages, including improved computational efficiency, enhanced visualization capabilities, and increased interpretability of the data.

The work of Mack Preston [4] also motivated us to explore more about the features being used from the CERT Dataset [1] to perform the various augmentation approaches.

In his thesis [4], Preston utilizes 100 features from the CERT dataset. He ranks the importance of these features in predicting the output class using Random Forest Regression SHAP (SHapley Additive exPlanations) [9], based on the Random Forest regression log-scaled importance. However, due to resource constraints, our work is limited to the R4.2 dataset.

Inspired by Preston's methodology, we aimed to extend his work by exploring a wider range of features from the CERT dataset. However, due to resource constraints, our investigation was limited to working specifically with the R4.2 dataset. Nonetheless, our goal was to extract maximum value from this subset of features to enhance our understanding of insider threat detection and mitigation.

By leveraging the R4.2 dataset, we were able to capture a diverse set of features that encompassed behavioral, temporal, and contextual aspects relevant to insider threat scenarios. This comprehensive feature set allowed us to analyze and model various dimensions of insider threats, providing a more holistic perspective on the problem.

While Preston's work focused on ranking features based on their importance, our approach involved utilizing all the available features to train and evaluate our deep learning models. We believed that considering the entirety of the feature space would enable us to capture the complex interplay between different variables and potentially uncover hidden patterns that could improve detection accuracy.

4. List of Approaches and Results

In this work, prior to training on the CERT dataset, we pre-processed using a technique proposed and implemented by Le *et al.* [10]. The technique involves the following steps:

1) **Grouping Log Events:** Events are parsed from log data and sorted into buckets based on user and time. Following Preston's work, we use daily time buckets as they were the best-performing time representation in prior works. [11].

2) **Feature Extraction:** Various permutations of statistical and count-based features are calculated for each day for each user.

3) **Temporal Representation:** A sliding window is passed over the extracted daily features for each user, computing the percentile of the current day with respect to the trailing window for the same user. We consider a window size of 30 days and the percentile metric due to its superior performance in prior studies. [11]

Insider threats pose a significant risk to organizations, and detecting and mitigating them requires advanced techniques such as classification machine learning models. In our research, we have implemented various classification models for insider threat detection, each with its specific parameters and ensemble techniques. This study aims to provide an in-depth exploration of these models, their

associated parameters, and the application of ensemble techniques to improve their performance.

Logistic Regression:

Logistic regression is a commonly used linear classification model that predicts the probability of an event occurring. In insider threat detection, logistic regression was utilized to analyze user behavior data and estimate the likelihood of an individual engaging in malicious activities. The key parameter in logistic regression is the regularization term, which controls the trade-off between model complexity and overfitting. Regularization parameters, such as L1 or L2 regularization coefficients, can be tuned to optimize model performance.

Decision Trees:

Decision trees are versatile classification models that use a tree-like structure to make decisions based on feature tests. The parameters of decision trees include the maximum depth, which determines the tree's depth, and the minimum number of samples required to split a node. By controlling these parameters, we can prevent overfitting and underfitting. Additionally, decision trees can be enhanced using ensemble techniques such as Random Forest or Gradient Boosting, which combine multiple decision trees to improve accuracy and robustness.

Random Forest:

Random Forest is an ensemble learning method that combines multiple decision trees. The key parameter in Random Forest is the number of trees in the ensemble, which affects model performance and computational efficiency. Additionally, Random Forest utilizes feature subsampling and sample subsampling. The feature subsampling parameter determines the number of features considered at each split, while the sample subsampling parameter controls the proportion of training instances used for each tree (Figure 3, Figure 4).

Support Vector Machines (SVM):

Support Vector Machines (SVMs) are powerful binary classification models that aim to find an optimal hyperplane that separates different classes. Parameters

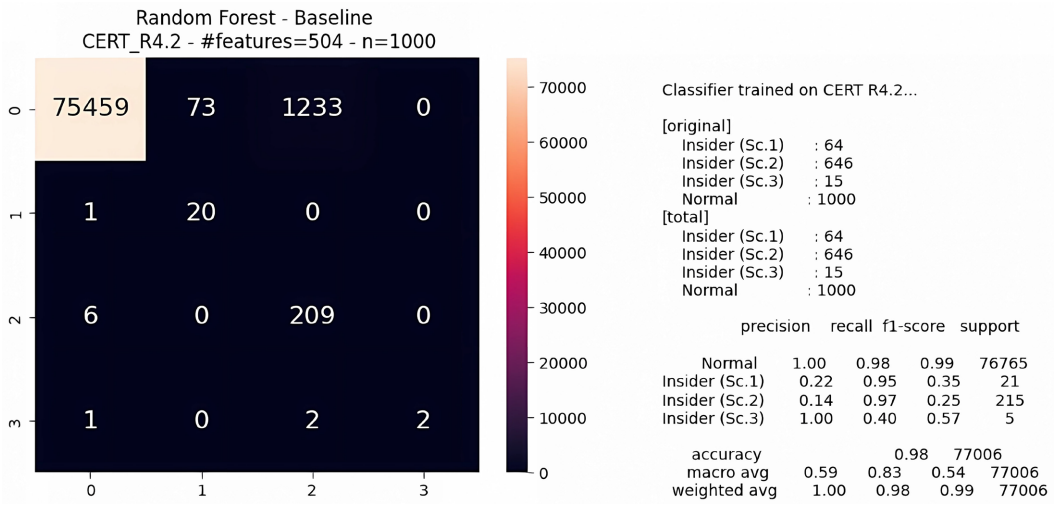

Figure 3. Results on Random Forest Regressor trained on r4.2 dataset.

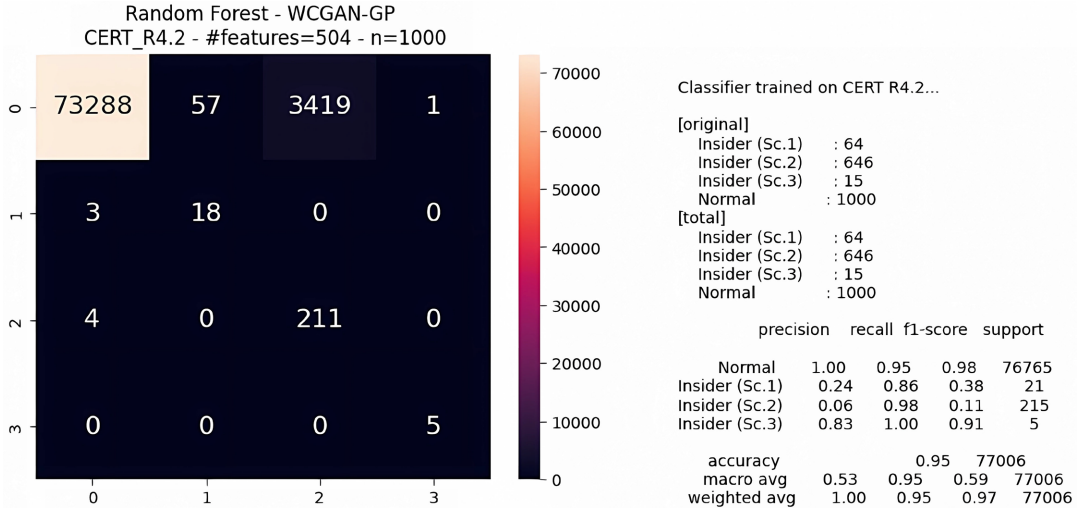

Figure 4. Results on Random Forest Regressor trained on r4.2 dataset + augmented data generated using WCGAN-GP.

in SVMs include the regularization parameter (C), which balances the importance of correct classification and maximizing the margin between classes. Another parameter is the kernel function, which determines the transformation applied to the data. Common kernel functions include linear, polynomial, and radial basis function (RBF).

Naive Bayes:

Naive Bayes classifiers are probabilistic models based on Bayes' theorem, assuming that features are conditionally independent. Although Naive Bayes models have a simple structure, they have proven to be efficient and effective in CERT classification tasks. The parameter in Naive Bayes models involves the smoothing factor (alpha or Laplace smoothing), which prevents zero probabilities.

Ensemble techniques:

Ensemble techniques involve combining multiple models or algorithms to produce a stronger and more robust prediction or classification. By leveraging the collective intelligence of diverse models, ensemble techniques aim to overcome the limitations of individual models and improve overall performance. Various ensemble techniques were utilized, including bagging, boosting, stacking, and random forests. Each technique employs a different strategy to generate a consensus decision based on the predictions of its constituent models.

Bagging:

Bagging, short for bootstrap aggregating, is an ensemble technique that involves training multiple models on different subsets of the training data. In the context of insider threat detection, bagging was applied to train multiple AI models using various subsets of historical user behavior data. By combining the predictions of these models, the ensemble system identifies patterns and anomalies in user behavior more accurately, reducing false positives and improving overall detection rates.

Boosting:

Boosting is another ensemble technique that sequentially trains multiple models, with each subsequent model focusing more on misclassified instances from previous models. In insider threat detection, boosting was employed to iteratively learn from misclassifications and refine the detection algorithm. By giving more weight to instances that were previously misclassified, boosting enables the ensemble system to adapt and improve its performance over time, increasing the accuracy of identifying insider threats.

Stacking:

Stacking is a technique that combines multiple models by training a meta-model on the predictions of the constituent models. In the context of insider threat detection, different AI models with distinct strengths and weaknesses were employed. These models generated individual predictions based on their unique features and algorithms. The meta-model then combines these predictions to make the final decision, leveraging the diverse insights of the constituent models. Stacking helped improve detection accuracy by considering a broader range of features and detection strategies.

General outcomes of using ensemble techniques for classification had higher accuracy than general machine learning models. The reason might be because ensemble techniques leverage the collective intelligence of multiple models, resulting in more accurate predictions and reducing false positives and false negatives in insider threat detection. By combining different models or algorithms, we found that ensemble techniques enhance the robustness of the detection system, ensuring that it can adapt to varying types of insider threats and changing attack patterns. Ensemble techniques like XGboost were able to incorporate diverse features and detection strategies, providing a comprehensive view of user behavior and enhancing the detection of sophisticated insider threats.

Generative models leverage statistical techniques to understand the underlying patterns and dependencies within the data, enabling them to generate new instances that closely resemble the original data. These models were trained on CERT datasets, learning the characteristics of normal behavior patterns and facilitating the identification of deviations or anomalies. Generative models can play a crucial role in insider threat detection by providing a comprehensive understanding of normal user behavior. By learning the statistical properties of legitimate activities, these models can identify deviations and abnormal behaviors that may indicate malicious intent.

Gaussian Mixture Models (GMMs):

Gaussian Mixture Models are widely used generative models that represent the probability distribution of the training data using a combination of Gaussian distributions. In the context of insider threat detection, GMMs were trained on historical user behavior data to learn the patterns and characteristics of normal activities. During detection, the model calculates the likelihood of a new behavior sequence belonging to the learned distribution.

Hidden Markov Models (HMMs):

Hidden Markov Models are probabilistic models that capture the underlying structure of sequential data. In the context of insider threat detection, HMMs can be utilized to model the sequence of user actions or events. The model was trained on normal user behavior sequences, and during detection, it calculates the likelihood of a new sequence given the learned model. If the likelihood falls below a certain threshold, it signifies an abnormal behavior pattern, potentially indicating an insider threat.

Variational Autoencoders (VAEs):

Variational Autoencoders (VAEs) are neural network-based generative models that learn a compact representation, or latent space, of the training data. VAEs are trained to encode and decode data, effectively learning the underlying distribution of the training data. In insider threat detection, VAEs were trained on normal user behavior data and subsequently used to reconstruct new behavior instances. Deviations or anomalies in the reconstructed instances can be identified as potential insider threats.

Generative models have helped us identify insider threats that exhibit subtle and previously unseen patterns by learning the normal behavior distribution and flagging deviations from it. These models have successfully captured the contextual dependencies between different user actions, providing a holistic view of normal behavior and enabling the detection of anomalous sequences or events. By modeling normal behavior patterns, generative models have helped reduce false positives by distinguishing legitimate deviations from malicious activities.

To enhance the accuracy and effectiveness of these systems, the choice of the right model is crucial. After extensive experimentation, Generative Adversarial Networks (GANs) and Wasserstein GANs (WCGANs) turned out to be the best models for AI insider threat detection. This study explores the reasons why GANs and WCGANs came out as the most suitable models for detecting insider threats.

By employing a generator and a discriminator network, GANs learn the underlying patterns and dependencies in the training data. This capability enables GANs to generate synthetic data samples that closely resemble the original distribution. In the context of insider threat detection, GANs can generate realistic behavior sequences, allowing for effective identification of anomalies and deviations from normal behavior patterns.

Insider threat detection often faces challenges related to limited labeled data and class imbalance, where instances of malicious activities are relatively scarce compared to normal behaviors. GANs can address these challenges by generating synthetic data samples to augment the training dataset.

GANs operate in an unsupervised learning setting, which has proven particularly advantageous for insider threat detection. Unsupervised learning allows GANs to learn the distribution of normal user behavior without relying on labeled instances of malicious activities. By learning the normal behavior patterns,

GANs can subsequently identify anomalies or deviations that fall outside this learned distribution. This ability to detect anomalies in an unsupervised manner makes GANs well-suited for identifying previously unseen insider threats and adapting to new attack patterns.

While GANs offer significant benefits for insider threat detection, they can be challenging to train. Traditional GANs often suffer from training instability, mode collapse, and vanishing gradients. Wasserstein GANs (WCGANs) address these issues by introducing a Wasserstein distance-based objective function, which provides more stable and reliable training. WCGANs mitigate mode collapse and allow for a smoother optimization process, resulting in better convergence and improved generation of realistic behavior samples. The stability and improved training of WCGANs make them highly suitable for insider threat detection, ensuring the model's performance is consistent and reliable. These properties combined prove that WCGAN is still the best approach among numerous traditional and ensemble techniques implemented before.

Adversarial training techniques can also be applied to GANs, where the generator and discriminator are simultaneously trained to outperform each other. This adversarial robustness enhances the model's ability to detect sophisticated insider threats that attempt to mimic normal user behavior.

5. Promising Approaches for Further Study

There can be numerous techniques introduced to the existing solution on AI insider threat detection. One of the most notable and simplest tasks to perform might be the inclusion of hyperparameter tuning for several other models. Ensemble techniques with hyperparameter tuning can significantly improve the overall performance by their ability to adaptively learn from weak learners. The effectiveness of an ensemble heavily relies on carefully selecting appropriate hyperparameters. Ensemble techniques, such as bagging, boosting, and stacking, combine the predictions of multiple base models to achieve better predictive accuracy and robustness. Each base model may have different hyperparameters that control its behavior and performance. Hyperparameters can include the number of base models, their types, learning rates, regularization parameters, and more. Properly configuring these hyperparameters is crucial for achieving optimal ensemble performance.

The first approach can involve the usage of grid search. Grid search is a commonly used technique that exhaustively searches through a predefined hyperparameter space. It evaluates the ensemble's performance for each combination of hyperparameters and selects the one that yields the best results. A more impactful technique might be Bayesian optimization. This is a sequential model-based optimization technique that leverages probabilistic models to find the optimal set of hyperparameters. It iteratively selects new hyperparameters based on the previous performance evaluations, aiming to maximize the ensemble's performance while minimizing the number of evaluations. Bayesian optimization is

particularly useful when the evaluation of each hyperparameter configuration is time-consuming or costly.

In addition to tuning the base models' hyperparameters, attention should also be given to ensemble-specific hyperparameters. These hyperparameters include the combination strategy (e.g., averaging, voting, weighted voting), the number of base models, and the diversity mechanisms (e.g., different algorithms, subsets of features). Properly tuning these ensemble-specific hyperparameters can have a significant impact on the overall performance of the ensemble. Automated hyperparameter tuning techniques, such as AutoML, can greatly simplify the process for tuning hyperparameters in ensemble techniques. AutoML tools employ sophisticated algorithms and optimization techniques to automatically search for and optimize hyperparameters. These tools can save time and effort by efficiently exploring the hyperparameter space and finding near-optimal configurations.

For example, one of the best machine learning models found during this study was XGboost. Implementing hyperparameter tuning for this model would involve using all possible combinations of its parameters. Parameters such as learning rate and number of boosting rounds can greatly impact the performance of the model, as they control the shrinkage and the number of boosting rounds, respectively.

Future research in AI insider threat detection should focus on improving the collection and preprocessing of relevant data. Insider threats are often subtle, making it crucial to capture nuanced behaviors and activities. Researchers can explore innovative methods to collect diverse data sources, including system logs, email communications, and social media activities. Additionally, developing efficient algorithms to preprocess and fuse data from different sources will enhance the quality and reliability of input for the WCGAN model.

While WCGANs have demonstrated promising results in detecting insider threats, future work should explore the integration of multiple detection techniques to enhance overall accuracy. Combining WCGANs with other AI models, such as recurrent neural networks (RNNs) or deep belief networks (DBNs), can improve the robustness of the system by capturing different aspects of insider threat behavior. Ensemble methods can be utilized to leverage the strengths of various models and mitigate individual weaknesses.

Providing contextual insights into detected insider threats is crucial for effective decision-making and an appropriate response. Future research should aim to develop techniques that not only identify potential threats but also provide explanations about why a particular behavior or activity was flagged as suspicious. This can be achieved by incorporating attention mechanisms or interpretable AI methods, enabling system administrators to understand the reasoning behind the alerts and take appropriate action accordingly.

Therefore, AI insider threat detection using WCGANs holds immense potential in mitigating the risks associated with insider threats. However, further research is required to improve data collection, adaptability to dynamic environ-

ments, integration with multiple techniques, contextual and explainable insights, privacy preservation, and real-world deployment. By addressing these areas, researchers can develop more robust and reliable systems that can effectively detect and mitigate insider threats, ultimately safeguarding organizational assets and maintaining data integrity.

6. Conclusions

The main purpose of this paper was to explore approaches to augment the CERT insider threat dataset using deep learning techniques and to experiment with the proposed methods. In this paper, we aim to test some of these experiments and build upon the works of Mack Preston, which we believe will enhance the detection and mitigation of insider threats in IT infrastructures. By utilizing deep learning techniques, specifically focusing on the use of Generative Adversarial Networks (GANs), our goal was to tackle the challenges of class imbalance and limited data availability in the CERT dataset.

Our experiments involved using all the features present in the CERT dataset to train and evaluate our models. This approach allowed us to capture a comprehensive representation of insider threat scenarios, including various behavioral, temporal, and contextual factors. By considering all available features, we aimed to enhance the robustness and accuracy of our models in detecting and predicting insider threats.

The utilization of all features in the CERT dataset did not significantly improve the performance of our models compared to previous approaches that focused on a subset of features. However, we found that the additional information provided by these features contributed to a more nuanced understanding of insider threat incidents and enabled better discrimination between normal behavior and malicious activities.

Moreover, our exploration of GANs as a means to augment the CERT dataset showed potential for generating synthetic insider threat instances that closely resemble real-world incidents. The diversity and adaptability of GAN-generated data allowed us to simulate different attack scenarios, insider profiles, and system conditions. This capability proved valuable in assessing the effectiveness of countermeasures and evaluating the resilience of security systems against evolving insider threat tactics.

While our experiments demonstrated the efficacy of utilizing all features and GAN-generated data, there are still important considerations and challenges to address. The ethical implications of using GANs to generate synthetic data and the potential biases introduced through the training process require careful attention. Additionally, further investigation is needed to determine the generalizability of our models to different organizational contexts and the scalability of GAN-based augmentation methods.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Glasser, J. and Lindauer, B. (2013) Bridging the Gap: A Pragmatic Approach to Generating Insider Threat Data. 2013 *IEEE Security and Privacy Workshops*, San Francisco, 23-24 May 2013, 98-104. https://doi.org/10.1109/SPW.2013.37

[2] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014) Generative Adversarial Networks. arXiv: 1406.2661. https://doi.org/10.48550/arXiv.1406.2661

[3] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A. (2017) Improved Training of Wasserstein GANs. arXiv: 1704.00028. https://doi.org/10.48550/arXiv.1704.00028

[4] Preston, M. (2022) Insider Threat Detection Data Augmentation Using WCGAN-GP. Master's Thesis, Dalhousie University, Halifax. https://library-archives.canada.ca/eng/services/services-libraries/theses/Pages/item.aspx?idNumber=1340918697

[5] Chawla, N.V., Bowyer, K., Hall, L. and Kegelmeyer, W. (2002) SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, **16**, 321-357. https://doi.org/10.1613/jair.953

[6] Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J. and Scholkopf, B. (1998) Support Vector Machines. *IEEE Intelligent Systems and Their Applications*, **13**, 18-28. https://doi.org/10.1109/5254.708428

[7] Patle, A. and Chouhan, D.S. (2013) SVM Kernel Functions for Classification. 2013 *International Conference on Advances in Technology and Engineering* (*ICATE*), Mumbai, 23-25 January 2013, 1-9. https://doi.org/10.1109/ICAdTE.2013.6524743

[8] Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., Saikhom, R., Panda, S. and Laishram, M. (2017) Multivariate Statistical Data Analysis—Principal Component Analysis (PCA). *International Journal of Livestock Research*, **7**, 60-78. https://doi.org/10.5455/ijlr.20170415115235

[9] Lundberg, S. and Lee, S.-L. (2017) A Unified Approach to Interpreting Model Predictions. 31*st Conference on Neural Information Processing Systems* (*NIPS* 2017), Long Beach, 4-9 December 2017, 1-10. https://www.researchgate.net/publication/317062430_A_Unified_Approach_to_Interpreting_Model_Predictions

[10] Le. D.C., Zincir-Heywood, N. and Heywood, M.I. (2020) Analyzing Data Granularity Levels for Insider Threat Detection Using Machine Learning. *IEEE Transactions on Network and Service Management*, **17**, 30-44. https://doi.org/10.1109/TNSM.2020.2967721

[11] Le, D.C. and Zincir-Heywood, N. (2020) Exploring Adversarial Properties of Insider Threat Detection. 2020 *IEEE Conference on Communications and Network Security* (*CNS*), Avignon, 29 June-1 July 2020, 1-9.