Generative Adversarial Networks (GAN) based Anomaly Detection in Industrial Software Systems

Tharindu Kumarage, Surangika Ranathunga, Chamal Kuruppu, Nadun De Silva, Malsha Ranawaka
Department of Computer Science and Engineering
University of Moratuwa
Katubedda, Sri Lanka
{sandaruwantk.13, surangika, chamalkuruppu.13, nadunrds.13, malsha.13}@cse.mrt.ac.lk

***Abstract*—Adopting an accurate anomaly detection mechanism is crucial for industrial software systems in order to prevent system outages that can deteriorate system availability. However, employing a supervised machine learning technique to detect anomalies in large production scale industrial software systems is highly impractical due to the requirement of annotated data. This raises the need for comprehensive semi-supervised and unsupervised anomaly detection mechanisms. This paper presents the application of Generative Adversarial Network (GAN) based models to detect system anomalies using semi-supervised one-class learning. We show that the use of a variant of GAN known as bidirectional GAN (BiGAN) gives augmented results when compared to the traditional GAN based anomaly detection, for the selected industrial system. Moreover, the experiments clearly show that the performance of the BiGAN has a direct correlation with the dimensions of the dataset used for training. The BiGAN even tends to outperform the well-established semi-supervised One-class SVM classifier and a prominent generative network for semi-supervised anomaly detection, Variational Autoencoders (VAEs) when the size of the feature space increases.**

***Keywords*—anomaly detection, industrial software systems, generative adversarial network,variational autoencoders, GAN, BiGAN, VAE**

I. Introduction

Due to the complex nature of the system components, manually monitoring or using a rule-based system for anomaly detection in industrial software systems is not feasible. Hence, machine learning based automated mechanisms for anomaly detection is widely used. In the current context, there are various anomaly detection mechanisms that come under the categories supervised, semi-supervised and unsupervised anomaly detection [1].

Even though supervised anomaly detection has shown superior results, these approaches lack flexibility and scalability, as they heavily depend on the domain of execution and the availability of annotated data. Furthermore, a supervised approach is not suitable for real-time learning models. Thus semi-supervised and unsupervised methods are more suitable for the problem at hand.

Many semi-supervised and unsupervised anomaly detection mechanisms such as clustering, one-class classification, and deep learning methods [2] have been used for anomaly detection in general. Out of the unsupervised deep learning techniques, various deep generative model-based techniques have performed well in anomaly detection [3]. However, there are various novel types of deep generative models that are yet to be employed for anomaly detection in the industrial software systems domain.

This paper introduces an anomaly detection framework based on a deep generative model known as Generative Adversarial Network (GAN) [4], [5] for anomaly detection in the industrial software systems domain. GANs are known for their power in image anomaly detection capabilities [5]. In particular, we employ a new variant of GAN known as Bidirectional GAN (BiGAN) [7], [8]. BiGAN adds more value to the practicality of the overall anomaly detection mechanism, since the inference machine (encoder) of the BiGAN can be immediately used to generate an anomaly score instead of the time-consuming calculation of the normal AnoGAN score method utilized in GAN [5].

The selected industrial software system for our experiments is a commercial stock trading software system [6]. For three datasets generated from three sub-components of this system, where each consisting of more than a hundred thousand data samples, we show that BiGAN outperforms the traditional GAN method.

We also discuss how the capabilities of generative models vary with the dimensions and number of training samples of the dataset. According to one of our previous work on this same industrial system, we showed that an Autoencoder based anomaly detection technique named Variational Autoencoders (VAEs) tend to give good accuracy values for the problem at hand [9]. However, in this current experiment, we evidently demonstrate that the performance of the BiGAN can outperform Variational Autoencoders for the component anomaly detection task when the data is high-dimensional. In fact, according to the best of our knowledge, this is the first such discussion that includes a comparison between VAE and BiGAN capabilities in the problem domain of anomaly detection. Furthermore, we show that BiGAN outperforms the well-known semi-supervised anomaly detection method, one-class SVM (Support Vector Machines), when the feature space is high-dimensional. Consequently, our results imply the potential of BiGAN to produce augmented accuracy rates in anomaly detection for datasets in the presence of a larger

978-1-7281-3632-5/19/$31.00 2019 IEEE

number of high-dimensional data samples.

The rest of this paper is organized as follows. Section II describes the literature review associated with the anomaly detection in industrial software systems, and section III defines the industrial software system used as the test bench. Section IV demonstrate the GAN based methodology for anomaly detection, and Section V comprises of a discussion on the experiments and evaluation, followed by the conclusion in Section VI.

II. Related Work

A. Anomaly Detection

An anomaly can be identified as a pattern in data that does not conform to the expected behavior [10]. Initial anomaly detection techniques were able to detect anomalies based on probabilistic feature correlation [11]. Moreover, there were some statistical models based on Hidden Markov model theories [12], [13], and Decision Trees [14]. Even though these methods showed successful results for small data requirements, they were not scalable due to the increasingly complex nature of system data. Furthermore, due to the undesirable time complexities of these algorithms, researchers opted for alternative supervised machine learning techniques such as genetic algorithms, fuzzy logic, Naive Bayes and Support Vector Machines (SVM) [15].

However, providing annotated data for supervised learning in a practical production environment is an infeasible process. Hence, many semi-supervised and unsupervised methods have been introduced. Among these techniques, one-class learning models have shown good performance as a semi-supervised anomaly detection technique. One-class SVM [22] and Autoencoders with one-class learning [23] can be classified as few of the well-known such techniques. Moreover, hybrid models that combine linear one-class SVM and deep neural networks have also performed well in large-scale anomaly detection problems [2].

Unsupervised anomaly detection techniques are based on the fundamental assumption that normal instances are far more frequent than anomalies in the test data. Self-Organized Maps (SOM) [24], K-means [27], and density-based clustering technique DBSCAN [28]–[30] had been employed over the years for unsupervised anomaly detection.

When considering unsupervised deep learning based anomaly detection, deep generative models have been utilized in the latest research. Autoencoder models are one of the most notable deep generative models that had been employed in applications such as network intrusion detection, pathological analysis, and financial fraud detection [3], [31]. There are variations of these Autoencoders such as Denoising Autoencoder [32] and Variational Autoencoder (VAE) [23] that have shown much higher results in anomaly detection than the traditional Autoencoders.

B. Anomaly Detection in Industrial Software Systems

When considering the industrial software system domain, many comprehensive methods have been executed so far to detect anomalies. When considering the supervised techniques, SVM classifier together with a feature selection method based on a tree-based ensemble model has been used in order to detect system anomalies of a trading system [6].

Moreover, there are some unsupervised anomaly detection schemes such as Gaussian mixture models [11] to characterize the probabilistic correlation between flow-intensities measured at multiple points of monitoring data of the system in order to detect anomalies in the system. Furthermore, there are some prototype online anomaly prediction systems that use only system-level metrics by integrating the 2-dependent Markov chain models with the tree-augmented Bayesian networks (TAN) model [33].

Some research was mainly based on the concept of self-healing systems, where the system is capable of predicting anomalies and preventing them even before the failure occurs. Here also Hidden Markov Models were used to heuristically identify the root cause of a fault in an unsupervised manner given the historical system feature data [13]. The same methodology was extended by another study, that leveraged Restricted Boltzmann Machines (RBMs) and contrastive divergence learning to analyze changes in historical data [34]. One of our previous work on anomaly detection in industrial software systems utilized a well-established generative network category known as Autoencoders for the semi-supervised and unsupervised anomaly detection [9]. Moreover, we were able to provide evidence that a variant of Autoencoder model known as Variational Autoencoder (VAE) tends to outperform state of the art semi-supervised one-class SVM and unsupervised DBSCAN based anomaly detection.

C. Generative Adversarial Network based anomaly detection

Generative Adversarial Networks (GAN) [4] is a new framework introduced for estimating deep generative models via an adversarial process, by the means of simultaneously trained two deep neural network models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. Here the entire system is trained using backpropagation.

GAN has been used to detect anomalies in medical imaging data as candidates for markers, which are relevant for disease progression and treatment monitoring [5]. There are many novel variations of the generative adversarial networks in the current literature. BiGAN is one such variation of GAN, fashioned by adding an inference model (encoder) into the training process [7], [8]. BiGAN based anomaly detection has been used to detect anomalies in publicly available datasets such as KDD and MNIST and has demonstrated good accuracy scores when compared to the direct GAN based anomaly detection and few other well-established semi-unsupervised techniques such as one-class SVM [35].

Fig. 1. Sub component communication architecture.

III. Industrial Software System

The industrial software system which is used in this research is a stock trading system and the communication architecture of the aforementioned subsystems is shown in Fig. 1.

Industrial software systems focused in this paper are systems that consist of multiple subsystems that interact with each other. A subsystem is a collection of multiple components that also have different tasks and shares a common goal that focuses on achieving the functionality of the subsystem.The subsystems and the components are defined by domain experts. Therefore this hierarchy of subsystems and components can have an arbitrary structure and the behavior of each component may differ from others.

The experiments are mainly focused on three main sub-components of the aforementioned system that were introduced in one of our previous work [9] namely, Matching Engine, Native Gateway, Sequencer. An anomaly occurring in this trading software system can be identified as an anomaly propagated from each one of the above-mentioned sub-components, thus independent component anomaly detectors were implemented to detect anomalies in the sub-component level. For testing, three data sets are used based on the components of the selected software system and each of these sub-components consists of a large number of features as shown in Table I.

IV. METHODOLOGY

A. Anomaly Detection System Overview

In our experiment, we adopt a component anomaly detection architecture that was presented by a previous anomaly detection work done on the same industrial trading software system [5]. This is shown in Fig. 2.

B. Generative Adversarial Network (GAN)

1) Structure of the GAN: A GAN consists of two major components as shown in the Fig. 3, which are Generator (G) and Discriminator (D).

1) Generator - This network generates data by forging the original data set.
2) Discriminator - This network outputs the probability of a generated data point (by Generator) belonging to the real data set.

2) Unsupervised Manifold Learning: The leaning procedure of a GAN is as follows. Given the training data X that conforms to the non-anomalous data of the system, the generator (G) learns the distribution P(g), which is to generate data points in the manifold X from a latent variable Z that conforms to a mapping G(z). During this process, the feedback of the discriminator will be taken into account by back propagating the Generator network with the discriminator's output decision on the generated data point. In this setting, the network architecture of the generator G is equivalent to a decoder and over the learning period, G will learn to generate(decode) a much closer data point to real ones of which the discriminator will not be able to distinguish the difference.

The discriminator model (D) of the GAN is basically an encoder neural network that maps a given data point to a single scalar value. This output scalar can be defined as a measure of probability for the credibility of the generators generating power of the data samples given the training data X. Thus, models D and G are simultaneously optimized through the two-player minimax game with value function V(G,D) [4] as shown by (1).

$$min_G max_{\ D} V(D,G) = \mathrm{E}_{x\sim Pdata(x)}[\log D(x)] + \mathrm{E}_{z\sim p_z(x)}[\log(1-D(G(z)))] \quad (1)$$

3) Latent space mapping of new data points: Provided a data point x, the target is to find a point z that is in the latent space, such that the data point G(z) is most similar to x and resides in the manifold X. While training the generator model learns the mapping G(z): z $\rightarrow$ x. However, one drawback of the traditional GAN model is that it does not contain an inference model to directly produce the inverse mapping $G^{'}$(x):x $\rightarrow$ z automatically.

TABLE I
Experimented components and feature dimensionality

Component	Dimension
Native Gateway	78
Sequencer	110
Matching Engine 1	357

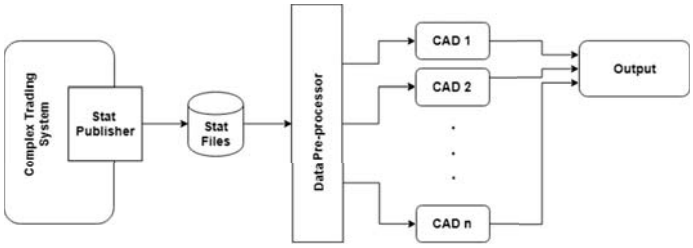

Fig. 2. Architecture of the component anomaly detection framework.

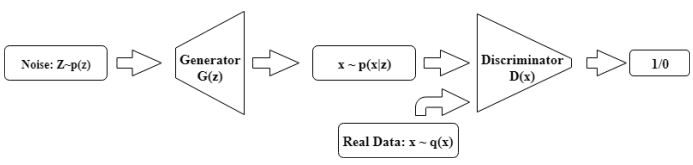

Fig. 3. Structure of the GAN.

Due to this reason, a random data input z is fed into the generator. This results in generating a data point G(z1). Based on the generated data point G(z1), a loss function that provides gradients for the update of the coefficients of z1 is defined, resulting in an updated position in the latent space, z2. The point in the latent space z2 generates a much closer data point to x. The location of z in the latent space Z is optimized in an iterative process by several back-propagation steps. Backpropagating the originally trained generator is not suitable in this case. Thus, another generator called 'inverse generator' is created for this task. Initially, this is a replica of the trained generator. After several iterations, this model generates a data point similar to x.

4) Anomaly detection procedure of the GAN: Anomaly score function of a GAN comprises of two main parts namely, residual loss and discrimination loss [5], [36].

1) Residual loss - Measures the dissimilarity between the given data point x and the generated data point G(z) in the data set. This is defined by (2)

$$L_R\left(Z_\gamma\right) = \sum |x - G(Z_\gamma)| \tag{2}$$

2) Discrimination loss- The probability of whether the generated data points lay inside the manifold of X.

Based on both residual loss and the discrimination loss, an anomaly score can be fashioned in such a way that larger the anomaly score greater the chance that the given data point to become an anomalous data point. Anomaly score A(x), as shown in equation (3) can be directly obtained from the residual loss R(x), discrimination loss D(x) and an anomaly weight μ, such that $0< \mu <1$.

$$A\left(x\right) = \left(1-\mu\right).R\left(x\right) + \ \mu.D(x) \tag{3}$$

C. Novel anomaly score calculation based on Bidirectional Generative Adversarial Network (BiGAN)

1) Structure of the BiGAN: In contrast to the structure of the normal GAN, an inference model is also added to the adversarial learning framework as shown in Fig. 4.

Similar to GAN, components of a BiGAN are also simultaneously optimized through a two-player minimax game with a modified value function V (G,D) [7,8] as shown by equation (4).

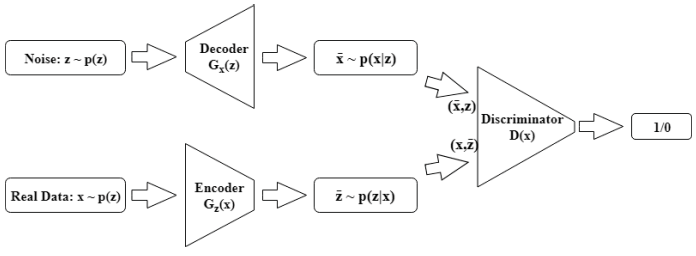

Fig. 4. Structure of the BiGAN.

$$\begin{aligned} min_G max\ _D V\left(D,G\right) = \mathrm{E}_{x\sim Pdata(x)}[\log D(x,G\left(x\right))] \\ +\mathrm{E}_{z\sim p_z(x)}[\log(1-D(G\left(z\right),z))] \end{aligned} \tag{4}$$

2) Anomaly detection procedure of the BiGAN: Similar to GAN, both residual loss and discrimination loss are utilized to create the anomaly score. However, in contrast to the traditional GAN, discrimination loss is calculated directly using the inference model (Encoder) as shown in equation (5).

$$L_D\left(Z_\gamma\right) = \sum |x - G_x(z)| \tag{5}$$

V. EXPERIMENTS

A. Dataset Generation

As mentioned in Section III, our experiment was based on a trading industrial software system, where we considered three predefined intercommunicating components to detect component-wise anomalies. Datasets for each component were separated into training and test sets. Moreover, the normal class was separated in order to train the GAN in an unsupervised manner using only the normal instances of data. The dataset generation process was adopted from a previous research experiment on the same industrial trading software system [6].

B. Experimental Results

The experiment conducted was three folds. As the first phase, both GAN and BiGAN anomaly detection mechanisms were tested with each of the three components to identify the best performing method. Then the BiGAN based anomaly detection method (the selected anomaly detection method from the first phase) was compared with the well-established one-class SVM and Variational Autoencoder anomaly detection techniques. As the third and last phase of the experiment, we evaluated the BiGAN performance with varying data sample sizes to understand its behaviour when the training data sample size is increased.

Performance of GAN and BiGAN on the dataset is given in Table II. From the results, it is clearly visible that the BiGAN outperforms the traditional AnoGAN approach for anomaly detection for every component of the industrial software system.

As the second phase, our experiment was extended to compare the BiGAN based anomaly detection with the well established semi-supervised anomaly detection using one-class SVM. This evaluation can be seen in Table III. Here, BiGAN outperforms the one-class SVM for the ME component of

TABLE II
GAN VS. BIGAN COMPARISON

Comp	Recall		Precision		F1	
	GAN	***BiGAN***	***GAN***	***BiGAN***	***GAN***	***BiGAN***
ME	0.37	0.66	0.40	0.43	0.38	0.52
SEQ	0.38	0.53	0.18	0.41	0.25	0.46
NG	0.54	0.63	0.20	0.23	0.29	0.34

the trading system. For the SE and NG components, one-class SVM has better accuracy values than the BiGAN based anomaly detection.

TABLE III
BIGAN VS. ONE-CLASS SVM COMPARISON

Comp	Recall		Precision		F1	
	BiGAN	***SVM***	***BiGAN***	***SVM***	***BiGAN***	***SVM***
ME	0.66	0.32	0.43	0.23	0.52	0.27
SEQ	0.53	0.72	0.41	0.36	0.46	0.48
NG	0.63	0.99	0.23	0.24	0.34	0.39

According to one of our previous work on this same industrial system, we got good accuracy values from unsupervised Variational Autoencoders (VAE) with supervised feature selection [9]. Table IV compares the BiGAN accuracy values with values got from this VAE and we can clearly see that BiGAN performs well with the ME component where the feature dimension is higher than the other two components.

C. Experiments on dataset size

In order to analyze the effect of the size of training data samples towards the performance of the BiGAN based anomaly detection, we extended the experiment by recording the accuracy measures for varying training dataset sample sizes. This evaluation is depicted in Figure 5.

D. Results analysis

When considering the first phase results shown in Table II, we can clearly see that BiGAN based novel anomaly score calculation method outperforms the traditional GAN approach.

TABLE IV
BIGAN VS. VAE COMPARISON

Comp	Recall		Precision		F1	
	BiGAN	***VAE***	***BiGAN***	***VAE***	***BiGAN***	***VAE***
ME	0.66	0.35	0.43	0.66	0.52	0.46
SEQ	0.53	0.59	0.41	0.56	0.46	0.57
NG	0.63	0.81	0.23	0.32	0.34	0.46

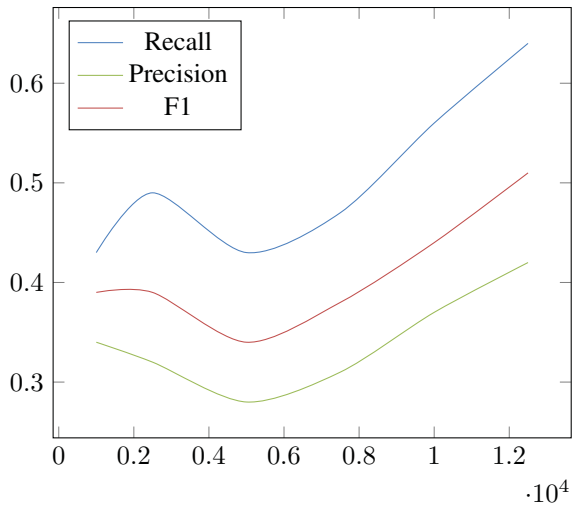

Fig. 5. Performance of BiGAN vs training dataset sample size.

This can be explained by the fact that BiGAN comprises an inbuilt encoder that maps input x to the latent model z which is optimized via the objective function at the training phase. However, in GAN we are creating the inverse mapping x to z separately after the training phase is over. Thus BiGAN model comprises a sound Encoder for input x to latent z mapping than the GAN model, which in fact affects the anomaly detection accuracy.

Moreover, BiGAN has outperformed the well-known one-class SVM for the experimental industrial system when the feature dimensions are higher. Thus we can perceive that BiGAN based anomaly detection tends to perform well with high-dimensional data. This is intuitive since higher the data dimensions, higher the complexity of the hypothesis that governs the anomaly detection. Therefore, BiGAN is perfectly capable of building a complex nonlinear hypothesis for the task of anomaly detection in high-dimensional datasets than the one-class SVM. This is further proven when considering the accuracy value comparison between BiGAN and VAE since both these models are capable of building non-linear complex hypotheses. However, when comparing BiGAN with VAE, adversarial learning of BiGAN helps to acquire a comprehensive latent model when it comes to the complex high-dimensional datasets than the variational inference used in VAE. Consequently, it can be seen that BiGAN outperforms VAE in ME component that consists of a larger feature set.

However, overall performance values are at a rate which is not quite suitable for a production environment. As a result, we extended our research in order to evaluate more deep into analyzing the performance. Through this empirical experiment, we found that the performance of the GAN is strongly correlated with the training dataset size. This evaluation is shown in Figure 5. Hence, it is justifiable to assume that with a training dataset consisting of a larger volume of a high-dimensional data sample, will give higher performance values for GAN based anomaly detection in this industrial system.

VI. CONCLUSION

This paper presented the application of a deep generative network known as a Generative Adversarial Network (GAN) for anomaly detection in an industrial software system that consists of components holding a large number of features. We showed that a new BiGAN based anomaly detection scheme performs better than the traditional GAN based anomaly detection (AnoGAN). BiGAN is more practical for a large-scale production environment due to its immediate anomaly score calculation scheme. Even though the results were quite below a recommended level, it is clearly seen that the number of samples used for training is highly positively correlated with the performance of the BiGAN. Moreover, it is evident that the BiGAN performs well among the sub-components that comprise of high feature dimensions. The aforementioned claim was further proven when comparing the accuracy values of BiGAN with the well-known semi-supervised anomaly detection approach, one-class SVM and also with our previous work on the same industrial system using VAEs. Thus, we

can conclude that with a higher number of data samples and higher feature dimensions for training, BiGAN will give relatively better results in anomaly detection in industrial software systems.

As future work, we intend to experiment on a test industrial system that consists of a larger production data sample set for training. Moreover, we are planning to experiment on anomaly detection using Adversarial Variational Bayes (AVB) networks, which is also known as Adversarial Autoencoders [37], a network which establishes a principled connection between GANs and Variational Autoencoders (VAEs). This way we can extract qualities of both networks into one unified scheme in order to detect anomalies in industrial software systems.

ACKNOWLEDGMENT

We would like to thank LSEG Technology, for providing the necessary datasets in order to evaluate the proposed anomaly detection methodology.

REFERENCES

[1] S. Agrawal and J. Agrawal, "Survey on Anomaly Detection using Data Mining Techniques," *Procedia Computer Science*, vol. 60, pp. 708-713, 2015.

[2] S. Erfani, S. Rajasegarar, S. Karunasekera and C. Leckie, "High-dimensional and Large-scale Anomaly Detection using a Linear One-class SVM with Deep Learning," *Pattern Recognition*, vol. 58, pp. 121-134, 2016.

[3] M. Sakurada, T. Yairi, "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction," in *Proceedings of the 2nd Workshop on Machine Learning for Sensory Data Analysis, MLSDA 2014*, pp. 4, 2014.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y.Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pp. 2672–2680, 2014.

[5] T. Schlegl, P. Seebock, S. M. Waldstein, U. Schmidt-Erfurth, G. Langs, "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery," in *Information Processing in Medical Imaging - 25th International Conference, IPMI*, pp. 146–157, 2017.

[6] L. Ranaweera, R. Vithanage, A. Dissanayake, C. Prabodha, S. Ranathunga, "Anomaly Detection in Complex Trading Systems," in *Engineering Research Conference (MERCon)*, pp. 437442, 2017.

[7] J. Donahue, P. Krähenbühl, T. Darrell, "Adversarial Feature Learning," *CoRR, abs/1605.09782*, 2016.

[8] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, A. Courville, "Adversarially Learned Inference," *CoRR. abs/1606.00704*, 2016.

[9] T. Kumarage, N. D. Silva, M.Ranawaka, C. Kuruppu, and S. Ranathunga, "Anomaly Detection in Industrial Software Systems - Using Variational Autoencoders," in *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods*, 2018.

[10] V. Chandola, A. Banerjee, V. Kumar, "Anomaly detection," *ACM Computing Surveys*, vol. 41, pp. 1-58, 2009.

[11] Z. Guo, G. Jiang, H. Chen, K. Yoshihira, "Tracking Probabilistic Correlation of Monitoring Data for Fault Detection in Complex Systems," in *International Conference on Dependable Systems and Networks (DSN'06)*, pp. 259-268, 2006.

[12] T. Ide, H. Kashima, "Eigenspace-based Anomaly Detection in Computer Systems," in *10th International conference on Knowledge discovery and data mining, SIGKDD*, pp. 440449, 2004.

[13] C. Schneider, A. Barker, S. Dobson, "Autonomous Fault Detection in Self-healing Systems: Comparing Hidden Markov Models and Artificial Neural Networks," in *International Workshop on Adaptive Self-tuning Computing Systems*, pp. 24, 2014.

[14] J. Alonso, L. Belanche, D. Avresky, "Predicting Software Anomalies using Machine Learning Techniques," in *10th IEEE International Symposium on Network Computing and Applications (NCA)*, pp. 163170, 2011.

[15] W. Hu, Y. Liao, V. Vemuri, "Robust Support Vector Machines for Anomaly Detection in Computer Security," in *ICMLA*, pp. 168174, 2003.

[16] P. Garca-Teodoro, J. Daz-Verdejo, G. Maciá-Fernández and E. Vázquez, "Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges," *Computers Security*, vol. 28, no. 1-2, pp. 18-28, 2009.

[17] S. Wang, "A Comprehensive Survey of Data Mining-based Accounting-fraud Detection Research," in *International Conference on Intelligent Computation Technology and Automation (ICICTA)*, pp. 5053, 2010.

[18] S. Padvekar, P. Kangane, K. Jadhav, "Credit Card Fraud Detection System," *International Journal Of Engineering And Computer Science*, 2016.

[19] P. Kanhere and H. K. Khanuja, "A Survey on Outlier Detection in Financial Transactions," *International Journal of Computer Applications*, vol. 108, no. 17, pp. 23-25, 2014.

[20] M. Ahmed, A. Mahmood and M. Islam, "A Survey of Anomaly Detection Techniques in Financial Domain," *Future Generation Computer Systems*, vol. 55, pp. 278-288, 2016.

[21] D. Delen, G. Walker and A. Kadam, "Predicting Breast Cancer Survivability: A Comparison of Three Data Mining Methods," *Artificial Intelligence in Medicine, vol. 34, no. 2*, pp. 113-127, 2005.

[22] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled data," *Applications of data mining in computer security*, vol. 6, pp. 77–102, 2002.

[23] J. An, S. Cho, "Variational Autoencoder based Anomaly Detection using Reconstruction Probability," in *Special Lecture on IE 2*, pp. 1-18, 2015.

[24] D. Dean, H. Nguyen, X. Gu, "Unsupervised Behavior Learning for Predicting Performance Anomalies in Virtualized Cloud Systems," in *9th International Conference on Autonomic Computing ICAC'12*, pp. 191-200, 2012.

[25] A. Jayasimhan, J. Gadge, "Anomaly Detection using a Clustering Technique," *International Journal of Applied Information Systems*, vol. 2, pp. 5-9, 2012.

[26] M. Amer, S. Abdennadher, "Comparison of Unsupervised Anomaly Detection Techniques," *Bachelor's Thesis*, 2011.

[27] R. Velea, C. Ciobanu, L. Margarit and I. Bica, "Network Traffic Anomaly Detection Using Shallow Packet Inspection and Parallel K-means Data Clustering," *Studies in Informatics and Control*, vol. 26, no. 4, 2017.

[28] P. Batra Nagpal, P. Ahlawat Mann, "Comparative Study of Density based Clustering Algorithms," *International Journal of Computer Applications*, vol. 27, pp. 44-47, 2011.

[29] H. Kriegel, P. Krger, J. Sander, A. Zimek, "Density-based Clustering. Wiley Interdisciplinary Reviews," *Data Mining and Knowledge Discovery*, vol. 1, pp. 231-240, 2011.

[30] Z. Chen, Y. Li, "Anomaly Detection Based on Enhanced DBScan Algorithm," *Procedia Engineering*, vol. 15, pp. 178-182, 2011.

[31] R. Aygun, A. Yavuz, "Network Anomaly Detection with Stochastically Improved Autoencoder based Models," in *4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 193198. 2017.

[32] P. Vincent, H. Larochelle, Y. Bengio, P. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," in *25th international conference on Machine learning*, pp. 10961103, 2008.

[33] Y. Tan, H. Nguyen, Z. Shen, X. Gu, C. Venkatramani, D. Rajan, "Prepare: Predictive Performance Anomaly Prevention for Virtualized Cloud Systems," in *32nd International Conference on Distributed Computing Systems (ICDCS)*, pp. 285294, 2012.

[34] C. Schneider, A. Barker, S. Dobson, "Autonomous Fault Detection in Self-Healing Systems using Restricted Boltzmann Machines," *CoRR. abs/1501.01501*, 2015.

[35] H. Zenati, C. Foo, B. Lecouat, G. Manek and V. Chandrasekhar, "Efficient GAN-based Anomaly Detection," *arXiv preprint arXiv:1802.06222*, 2018.

[36] R. Yeh, C. Chen, L. Chen, Y. Teck, M. Hasegawa-Johnson, D. Mark, N. Minh, "Semantic Image Inpainting with Perceptual and Contextual losses," *arXiv preprint arXiv, 1607.07539*, 2016.

[37] A. Creswell and A. A. Bharath, "Denoising Adversarial Autoencoders," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 117, 2018.