# Detection of DDoS Attacks Using Variational Autoencoder-Based Deep Neural Network

**Chapter** · October 2023

DOI: 10.4018/979-8-3693-0593-5.ch017

Agripah Kandiero
Africa University
**16** PUBLICATIONS **10** CITATIONS

Panashe Chiurunge
Chinhoyi University of Technology
**9** PUBLICATIONS **12** CITATIONS

# Chapter 17

# Detection of DDoS Attacks Using Variational Autoencoder-Based Deep Neural Network

**Agripah Kandiero**
https://orcid.org/0000-0001-8201-864X
*Instituto Superior Mutasa, Mozambique & Africa University, Zimbabwe*

**Panashe Chiurunge**
*Chinhoyi University of Technology, Zimbabwe*

**Jacob Munodawafa**
*University of St Thomas, Mozambique*

## ABSTRACT

*Distributed denial of service (DDoS) attacks are one of the most commonly used tools to disrupt web services. DDoS is used by groups of diverse backgrounds with diverse motives. To counter DDoS, machine learning-based detection systems have been developed. Proposed is a variational autoencoder (VAE) based deep neural network (VAE-DNN) classifier that can be trained on an unbalanced dataset without needing feature engineering. A variational autoencoder is a type of deep neural network that learns the underlying distribution of computer network flows and models how the benign and DDoS classes were generated. Because a VAE model learns the distribution of the classes within the dataset, it also learns how to separate them. The variational autoencoder-based classifier can scale to any data size. A deep neural network, quadratic discriminant analysis (QDA), and linear discriminant analysis (LDA) decision boundaries are applied to the latent representation of network traffic to classify the flows. The DNN shows the highest precision and recall of the three classifiers.*

DOI: 10.4018/979-8-3693-0593-5.ch017

# 1. INTRODUCTION

## 1.1 Background of the Research Problem

Malicious actors and their activities have found space on the internet. The use of the internet to access services of all kinds has also seen malicious actors bent on disrupting the services. All organisations that offer web-based services are vulnerable to DDoS attacks. Financial services, commercial, news, politics and entertainment, are now accessible through the internet. Malicious actors with diverse motives target web-based servers intending to bring down the services or at least degrade the quality of service offered to genuine customers. One of the most common ways of disrupting a web-based service is Distributed Denial of Service (DDoS). The malicious actors attempt to bring down or at least slow down a service by overwhelming the server's resource limits in terms of CPU, memory or network bandwidth. The attackers launch DDoS attacks by taking advantage one way or the other of protocols found at the Application layer, Transport layer and Network layer of the TCP/IP protocol suite. All network communication devices implement the TCP/IP protocols when they communicate. The Internet of Things has dramatically increased the number of online devices. The number of devices that can be recruited to form botnets which are then used to amplify DDOS attacks is huge (Kim, 2020). The TCP/IP model together with the OSI models, were not built with security in mind. When these models were proposed the motivations for most current cyber-attacks were not as prevalent as they are today since there was very little online activity then. The majority of Network capable devices such are those found among IoT terminals were also not built with the required focus on security. These appliances are, therefore, easily targeted and recruited to increase the size of botnets. Effective network intrusion detection systems are needed to counter the many possible DDOS attack vectors.

Malicious actors have various ways of implementing DDoS attacks. A botnet or network of compromised remotely controlled computers is commonly used to launch DDoS attacks (Khan, 2019). Other malicious actors amplify their DDoS attacks by making use of computers on various networks broadcast domains to send packets to a server at the same time to choke the server's processing capacity and resources. The malicious actors evolve their mechanisms of attack as technology evolves. Early botnets used Internet Relay Chat (IRC) protocol for sending instructions to the botnets from the botmaster using a client-server networking model. Then peer-to-peer (P2P) protocol was used by botnets before HTTP became the most commonly used protocol (Kim, 2020). In 2016 a botnet called Mirai took over control of hundreds of thousands of Internet of Things (IoT) devices that resulted in a widespread loss of Internet access in the USA. The botnet managed to recruit a large number of Internet of Things (IoT) devices that used default login credentials for Telnet protocol. On 29 January 2018, three Dutch banks ABN AMRO, ING Bank and Rabobank were attacked by DDoS resulting in loss of internet banking and inaccessibility of websites. In 2016 Twitter, Sound Cloud, Spotify, and Shopify all went down after a DDoS attack was launched against the cloud domain hosting company Dynamic DNS (Bonguet, 2017). With DDoS defence and mitigation systems, time is limited from the launch of attack to mitigation (Wangy, 2017).

DDoS attacks cause loss of revenue to service providers and cause inconveniences and frustration to customers as services become inaccessible. A DDoS detection and mitigation strategy is called for as part of a cybersecurity strategy. Cybersecurity focuses on preventing, detecting and reacting to threats and attacks timeously (Darko Galinec, 2017). The research offers a solution for detecting DDoS attacks. The built system can be adapted to be part of a broader cybersecurity strategy. Since the launching of DDoS attacks cannot be prevented, detecting DDoS timeously with high precision is the best that can be done. However, for high precision models to be trained, the dataset needs to be balanced. For all network intrusion detection and DDoS attack detection, correctly balanced datasets, are hard to get. Some techniques to balance the dataset before model training have been explored. The methods that have been adopted to balance data sets which include under-sampling the majority class and Oversampling of the minority class are discussed in Section 2 Literature Review. It is also essential that the DDoS detection systems be scalable to train models that can handle extensive network traffic data associated with the Internet and the evolving Internet of Things. Traditional machine learning classifiers also fell short of scalability besides requiring that the data set be balanced before training the model. Currently adopted data balancing techniques will be reviewed and show how the VAE offers an alternative to deal with unbalanced data sets. The scalability that cannot be achieved by traditional machine learning techniques is achieved by the VAE. VAE achieves scalability because it uses a mini-batch Stochastic Gradient when updating weights during training. The re-parameterization trick which makes stochastic gradient possible with VAE is explained in Section 2 Literature Review. The continually evolving nature of DDoS attacks is addressed by the profound neural network nature of A Variational Autoencoder, which makes a VAE automatically learn new patterns in network flows. Reliable datasets usable for training intrusion detection models are hard to get. In this research, the CICIDS 2017 dataset is used to train a VAE DDoS detector.

## 1.2 Research Problem

Distributed Denial of Service (DDOS) attacks result in loss of revenue and frustrated users as servers fail to serve legitimate users of web-based services. Timeous Detection of DDoS with high precision and high recall, therefore, saves revenue and guarantees customer satisfaction. However traditional machine learning algorithms (Support Vector Machines, Logistic Decision Trees, Logistic Regression, Naïve Bayes) which have been applied to detect DDoS attacks cannot train accurate classifiers if the training data set has an odd number of tuples for each class (Bellinger, 2017). In real-world situations, balanced data sets are rare. In many other applications, the data set may contain only the standard samples and have no anomalous samples for the model to learn from. Unbalanced data sets and lack of minority class samples in the dataset are a big challenge for DDoS detection systems and all intrusion detection systems in general. Although deep neural network classifiers perform better than traditional machine learning classifiers, their performance is also limited if the training data set is unbalanced data (Gupta, 2019). Dataset balancing techniques based on oversampling the minority class or under-sampling the majority class have been explored. Samples generated by oversampling of the minority class result in overfitting and samples generated by oversampling do not bring new knowledge for the model to learn. Classifiers trained on datasets balanced by sampling techniques showed some slight performance improvement. For training a classifier with an unbalanced dataset and still achieving high precision and high recall, a Variational Autoencoder (VAE) is proposed. For the very high precision and recall called for by DDoS detection systems and all intrusion detection systems, a Variational Autoencoder is proposed. A VAE is a deep generative model that models how a dataset is generated by learning the underlying distribution of the dataset.

VAE has the advantage of scalability over traditional machine learning classifiers. Like conventional deep neural networks, VAE models are scalable to any data set size because VAE also uses Mini-batch Stochastic Gradient Descent (SGD). Mini-batch SGD is a type of Stochastic Gradient Descent (SGD) which is used to update weights during training. VAE uses mini-batch SGD although the nature of the latent layer is random and probabilistic (Kurien, 2019). With mini-batch SGD, the gradient is calculated after each mini-batch is run. The VAE takes in a small batch of training samples for each iteration of loss minimization. Mini-batch SGD results in faster training of the model than Stochastic Gradient and also allows for parallel computations to speed up the training. For mini-batch SGD to be possible, the re-parameterization trick is used for backpropagation to go around the random latent space when updating VAE weights during training. Traditional data set balancing techniques such as oversampling of the minority class and under-sampling of the majority class can be made redundant by the use of Variational Autoencoders. VAE make training classifiers direct without prior extensive data preparation, such as balancing the data sets. With minimal modification, the VAE can generate new minority samples to balance a dataset. This research will demonstrate how a DDoS detection system can be built without the need for balancing the data set and extensive feature engineering. The VAE is coupled to two downstream classifiers which are a Deep Neural Network and Quadratic Discriminant Analysis for detecting DDoS attacks.

## 1.3 Research Objectives

1. To build a Variational Autoencoder based Deep Neural Network that detects DDOS attacks with high precision and high recall.
2. To design and build a Variational Autoencoder model which does not require the balancing of the training data set.
3. To design and build a Deep Neural Network DDoS classifier whose input is the VAE latent representation.
4. To apply a Linear Discriminant Analysis decision boundary to the latent layer representation of network traffic data and separate the classes.
5. To apply the Quadratic Discriminant Analysis decision boundary to the latent layer representation of network traffic data and separate the classes

## 1.4 Conceptual Framework

### 1.4.1 Why Variational Autoencoder Based Classification

VAE was chosen to implement a DDoS detection because a VAE does not require a balanced data set for training the model and also because the VAE models can scale to any training data size. A VAE models how the data was generated. A VAE is one type of generative model discussed in section section 2 Literature Review. A VAE is a combination of two neural networks. The first neural network is the encoder which takes in data inputs and represents the data in a latent space with reduced dimensions. The latent space is in the form of a joint probability distribution of the essential variables only. The distribution is forced to be normal by minimization of a Kullback-Leibler (KL) divergence term. The second neural network takes samples from the latent space as input and attempts to reconstruct the original input data. The difference or dissimilarity between the reconstructed output and the original input is a loss to be minimized. During training, the loss function of the VAE minimizes the sum of the KL divergence loss and the reconstruction loss. A VAE attempts to recreate the given input while under constraints such as smaller hidden layers and the forcing of the joint probability distribution of features to be Gaussian.

The scalability comes from VAE's ability to use Stochastic Gradient Descent (SGD) despite having randomness along the path of backpropagation. The mathematical foundation of the VAE re-parameterization trick that makes SGD possible in VAE is discussed in Section 2 Literature Review. Re-parameterization externalizes the randomness of the latent space and represents each probabilistic distribution of a data point with a deterministic value which is necessary for updating weights during backpropagation. The VAE learns the distribution of features within the benign flows and the distribution of features within the DDoS flows. The VAE then represents the network flows as a continuous probability distribution. The benign and DDoS attack samples occupy distinct regions of the probability distribution. The patterns of normal, benign network flows and the DDoS network flows are used to train the classifier. A Supervised machine learning approach is adopted to train a VAE classifier that separates computer network traffic flows into DDoS and benign traffic. The trained VAE takes in the seventy seven features of the CICIDS 2017 data set representing a network flow. The VAE outputs a mean and a standard deviation indicating the region in the joint probability distribution where the network flow is likely to be found. A QDA, LDA and DNN are then trained on the VAE. The QDA and LDA apply quadratic and Linear decision boundaries respectively to the network flows and separate the benign and DDoS flows. A DNN is also trained on the VAE latent representation. A DDoS Classifier is built having the foundation of a VAE, with DNN as a downstream classifier. It is either a QDA or LDA because only the better classifier between them will be co-opted into the hybrid system. The DNN is essential in the hybrid system because of its scalability.

Publicly available DDoS detection data sets have unbalanced classes. This calls for the datasets to be balanced first before a model can be trained (Yilmaz, 2019). The VAE is trained with the CICIDS 2017 dataset. The CICIDS 2017 Friday Afternoon data set is unbalanced in favour of DDoS attacks. The data set contains tuples of benign and DDoS samples. It has a total of 225745 tuples of which 97718 were benign and 12827 were DDoS attack flows. Currently, used data balancing techniques generate synthetic samples that either cause overfitting or loss of information. Feature engineering demands time and domain expertise. A VAE based DDoS classifier is built without balancing the data nor particular choosing of features. Because no feature engineering is conducted, a VAE can be used where domain knowledge is limited.

### 1.4.2 CICIDS 2017 Data Set Features

The CICIDS 2017 Friday Afternoon data set has a total of 78 features as shown in Table 1. The CICIDS2017 data set was generated in a testbed network environment with DDoS attacks being generated on the afternoon of July the 7th 2017. The CICFlowMeter was used for extracting network traffic features. Features that were likely to be the most explanatory for DDoS were identified. For detecting DDoS attacks 'backward packet length', 'average packet size' and 'inter arrival time standard deviation' were selected. Seven common traditional machine learning classification algorithms were used to examine the performance and accuracy of the selected features (Sharafaldin, 2018).

A comparison is made between the results of the VAE based hybrid classifiers and those of K-Nearest Neighbours (KNN), Random Forest (RF), ID3, Adaboost, Multilayer perceptron (MLP), Naive-Bayes (NB), Quadratic Discriminant Analysis (QDA) as shown in Table 2. Precision and recall are the metrics to be used for the comparison because both False Positives and False Negatives need to be minimized. Accuracy cannot be used as a metric to detect DDoS attacks because the CICIDS2017 Friday Afternoon data set is not balanced. For DDoS attacks testbed environment DDoS packets may not be very different from real world DDoS attack packets because similar packet generation methods are usually employed. Table 1 shows the features that collectively determine whether a flow is benign or an attack in the CICIDS2017 data set.

*Table 1. CICIDS 2017 Friday afternoon data set features*

| Variable | Variable | Variable | Variable |
|---|---|---|---|
| Source IP address | Flow Inter-Arrival-Time Min | Packet Length Standard Deviation | Subflow Forwarding Packets |
| Destination IP address | Forwarding Inter-Arrival-Time Total | Packet Length Variance | Subflow Forwarding Bytes |
| Destination Port for TCP and UDP | Forwarding Inter-Arrival-Time Mean | FIN Flag Count | Subflow Backwarding Packets |
| Protocol IP protocol | Forwarding Inter-Arrival-Time Standard Deviation | SYN Flag Count | Subflow Backwarding Bytes |
| Flow Duration | Forwarding Inter-Arrival-Time Max | RST Flag Count | Initial Forwarding Win Bytes |
| Total Forwarding Packets | Forwarding Inter-Arrival-Time Min | PSH Flag Count | Initial Backwarding Win Bytes |
| Total Backwarding Packets | Backwarding Inter-Arrival-Time Total | ACK Flag Count | Forwarding Act Data Packets |
| Total Length of Forwarding Packets | Backwarding Inter-Arrival-Time Mean | URG Flag Count | Forwarding Segment Size Min |
| Total Length of Backwarding Packets | Backwarding Inter-Arrival-Time Standard Deviation | CWE Flag Count | Active Mean |
| Forwarding packet Length Maximum | Backwarding Inter-Arrival-Time Max | ECE Flag Count | Active Standard Deviation |
| Forwarding Packet Length Minimum | Backwarding Inter-Arrival-Time Minimum | Down/Up Ratio | Active Max |
| Forwarding Packet Length Mean | Forwarding PSH Flags | Packet Length Size Average | Active Min |
| Forwarding Packet Length Standard Deviation | Backwarding PSH Flags | Forwarding Segment Size Average | Idle Mean |
| Backwarding Packet Length Maximum | Forwarding URG Flags | Backwarding Segment Size Average | Idle Standard Deviation |
| Backwarding packet Length Minimum | Backwarding URG Flags | Forwarding Bytes Average | Idle Maximum |
| Backwarding packet Length Mean | Forwarding Header Length | Forwarding Packets Average | Idle Minimum |
| Backwarding packet Length Standard Deviation | Backwarding Header Length | Forwarding Bulk Rate Average | Label |
| Flow Inter-Arrival Time Mean | Packet Length Minimum | Backwarding Bytes Average | |
| Flow Inter-Arrival Time Standard Deviation | Packet Length Maximum | Backwarding Packets Average | |
| Flow Inter-Arrival Time Maximum | Packet Length Mean | Backwarding Bulk Rate Average | |

### 1.4.3 Features Selected For DDoS Detection

Sharafaldin et al identified a set of features whose parameters are most likely to determine whether a sample was a DDoS attack, as shown in Table 2. The first packet sent between two communicating devices determines the forward and backward directions of the flows. The forward direction is the source destination of the first packet and the backward direction is destination-source response to the first packet. The standard deviation size of the packet in the backward direction, Standard deviation time between two packets sent in the flow, Average size of the packet and Duration of the flow in microseconds were identified as features with parameters most determining if a sample is a DDoS attack.

*Table 2. Selected features for DDoS attack (Sharafaldin, 2018)*

| Label | Feature |
|---|---|
| DDoS | Backwarding Packet Length Standard Deviation |
| | Flow Duration |
| | Average Packet Size |
| | Flow Inter-Arrival-Time Standard Deviation |

*Note*. Adapted from 'Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization', by I.Sharafaldin and A.Lashkari and A.Ghorbani, 2018, Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), p. 108-116.

### 1.4.4 Metrics for VAE-Based DDoS Detection Classifiers

Accuracy cannot be used as a reliable metric when training a model for DDoS detection. Supervised Machine learning classifiers trained with imbalanced datasets are biased towards the majority class and against the minority class because they learn more from the majority samples. If a classifier is trained with a data set with 97% of the samples being the benign network flows and 3% being the malicious network flows, the classifier will achieve a 97% accuracy rating if the model guesses that every data point is a benign flow during evaluation. Using accuracy is only suitable for symmetric data sets where the class distribution is equal for the classes and the cost of false positives and false negatives are roughly the same. Therefore accuracy cannot be used as a metric for models trained with imbalanced dataset (Alkasassbeh, 2016). Using VAE based classifiers this research will demonstrate how a system with both high precision and high recall is achievable. By first learning the distribution of the dataset the Variational Autoencoder will be able to classify network traffic with high precision and recall. Network traffic may easily reach big data scale given the Internet of Things and forecasts on the total number of devices connected. A VAE is scalable to big data scales because it uses mini-batch stochastic gradient descent during training. During training the VAE learns to represent the data in a new latent representation where separation is enhanced. The VAE encoder gives the log probability of latent representation z given input data point x, that is $p(z/x)$. The VAE decoder samples from the learnt distribution to give the probability of reconstructing $x$ from the latent representation $p(x/z)$. After the VAE is trained, both a linear and quadratic decision boundary are applied to the latent variables to separate the two classes of benign and DDoS network traffic. A decision boundary that classifies network flows better implies that the data set fulfils the assumptions of the type of Discriminant Analysis system. A DNN model is also trained on the latent representation of a trained VAE. This model serves to compare and endorse the results of the Discriminant Analysis classifier. The results of the VAE based classifiers are compared against themselves and also compared to the results of traditional machine learning classifiers obtained on the same CICIDS 2017 data set when it was generated by Sharafaldin et al., 2018.

## 2. LITERATURE REVIEW

### 2.1 Introduction

Literature about how to build a DDoS Detection system with high precision and high recall is reviewed in this section. Previous Studies on how traditional machine learning algorithms dealt with data imbalances in intrusion DDoS and intrusion detection are covered. How the VAE, a generative model, takes away the need for balancing data set classes and feature engineering for downstream classifiers is discussed in section 2.7. This section refers to journals and other scholarly literature that cover studies and approaches to training adequate DDoS detectors. The assumptions that go along with Quadratic Discriminant Analysis (QDA) and Linear Discriminant Analysis (LDA) and their implication on classification performance of the VAE based classifier are covered later in the section. The types of DDoS attacks and how they take advantage of the protocols of the TCP/IP suite are covered next.

### 2.2 Definition of Critical Terms

**Variational Autoencoder**: A deep generative model that encodes data inputs into a probability distribution. The probability distribution is sampled by the decoder to attempt to reconstruct the original input.

**Re-parameterization trick**: The externalization of the randomness of the latent space of a Variational Autoencoder and represents each probabilistic distribution of each data point with a deterministic value for updating weights during backpropagation

**Latent space**: The representation of data points by a probability distribution in which the data points lie,

**Encoder**: Part of a Variational Autoencoder that takes in data inputs and transforms the input into a probability distribution mean and standard deviation that represents the position of the data point in the distribution.

**Decoder**: Part of the Variational Autoencoder that samples from the latent space and attempts to reconstruct original input.

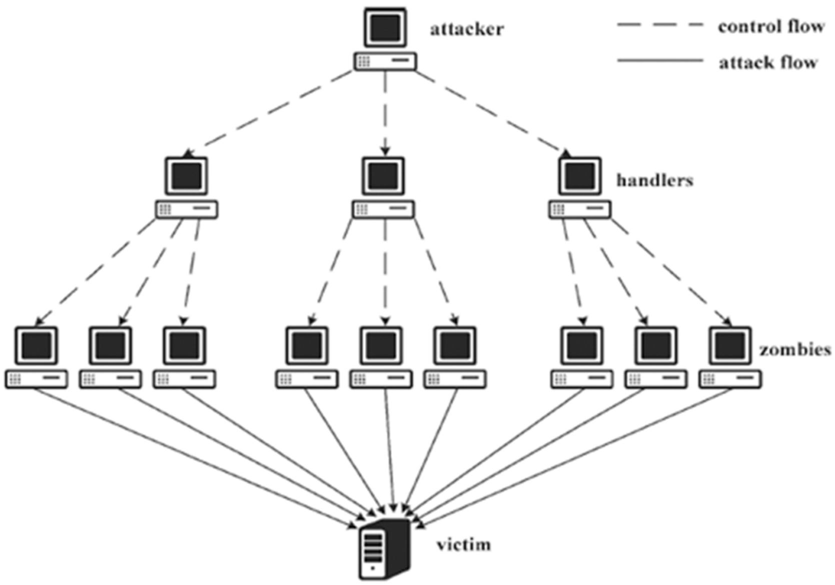
### 2.3 Types of DDoS Attacks

There are many forms of DDoS attacks. All the different attacks aim to deny genuine users access to a web-based service. The attacks take advantage of how the different layers of the TCP/IP suite implement protocols (Macfarlane, 2015). At the application layer, there are attacks such as HTTP flooding. At the transport layer, there are SYN flooding and UDP flooding. At the network layer, there are smurf attack ICMP flooding. The motivation for carrying out DDoS attacks could be varied since those who carry them come from different backgrounds such as social activism, political and outright criminal (Macfarlane, 2015). Since there is a demonstrated desire to commit distributed denial of service attacks on varying web services by a diverse group of people, it is imperative that intrusion detection systems which can detect the DDoS effectively be implemented. Common types of DDoS attacks are discussed next.

### 2.3.1 Botnet

Botnets are recruited to amplify and make attacks come from multiple sources. A DoS attack uses one computer and one Internet connection to flood a targeted system or resource. A botnet uses many computers to launch a coordinated DoS attack against one or more targets. A specific strategy is used to recruit, exploit, and infect computers (Alzahrani, 2018). During the recruiting phase, the attacker identifies and groups several machines from which to generate huge volumes of traffic to flood the victim server. The recruiting phase is accomplished through the scanning of remote machines using penetration testing software tools such as Nmap looking for security vulnerabilities. During The exploit phase, the attacker exploits the identified vulnerabilities to break into the recruited machines. The attacker infects the recruited machines with a DDoS program during the infect phase. A DDoS master program is installed on one computer using a stolen account. The master program, at a designated time, then controls and commands the many agent programs, installed on recruited computers. In response to a command from the botmaster, the agents initiate the attack. The infected machines are remotely controlled to launch the attack by the attacker. In this way, the perpetrator can multiply the effectiveness of the Denial of Service significantly by harnessing the resources of multiple compromised accomplice computers which serve as attack platforms. The master program can initiate thousands of agent programs in a short time. Figure 1 shows the underlying architecture of a botnet network.

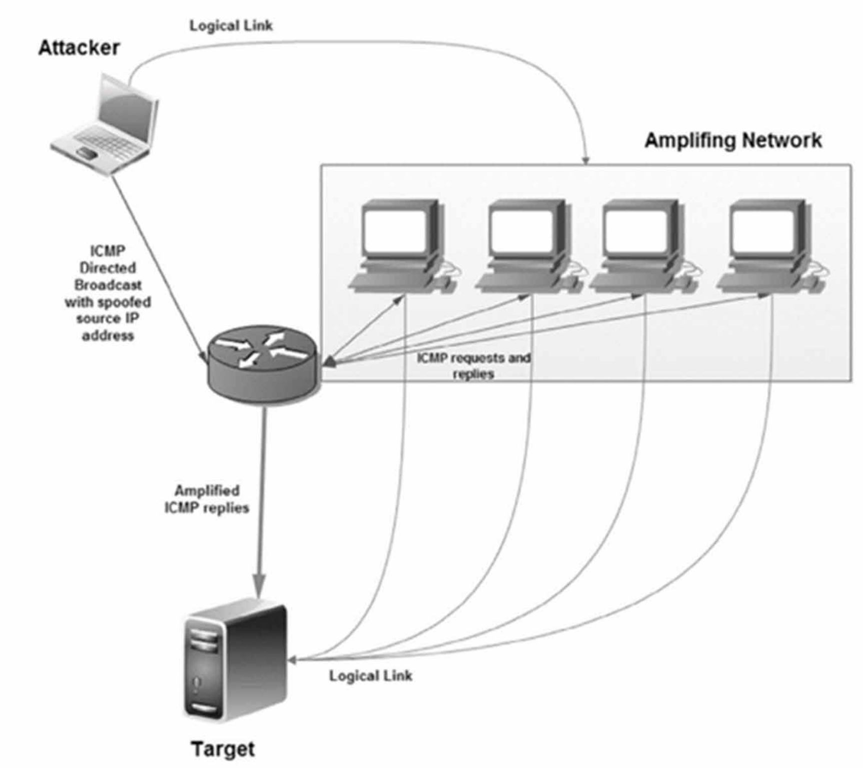
*Figure 1. Botnet DDoS attack architecture (Singh, 2015)*



### 2.3.2 Smurf Attacks

A smurf attack is a type of network layer DDoS which takes advantage of how the ICMP accomplishes its tasks. An attacker floods a targeted server with Internet Control Message Protocol (ICMP) packets. Requests are made to computer networks using the spoofed source IP address of the targeted device, as shown in Figure 2. The computer networks then sent responses to the targeted server thereby amplifying the initial attack traffic overwhelming the victim and making it unusable by legitimate users. To deplete the bandwidth of the targeted server, a smurf attack usually broadcasts messages whose replies will all be sent to the spoofed IP address of the victim. Every machine in the broadcast domain which received the broadcast then sends ICMP ECHO reply packets to the victim IP address (Mitra, 2017). The victim will not be able to deal with the sudden rise in number of reply messages. During the time of the attack, the victim server will fail to service requests from its authentic users or may give degraded service or crush altogether.

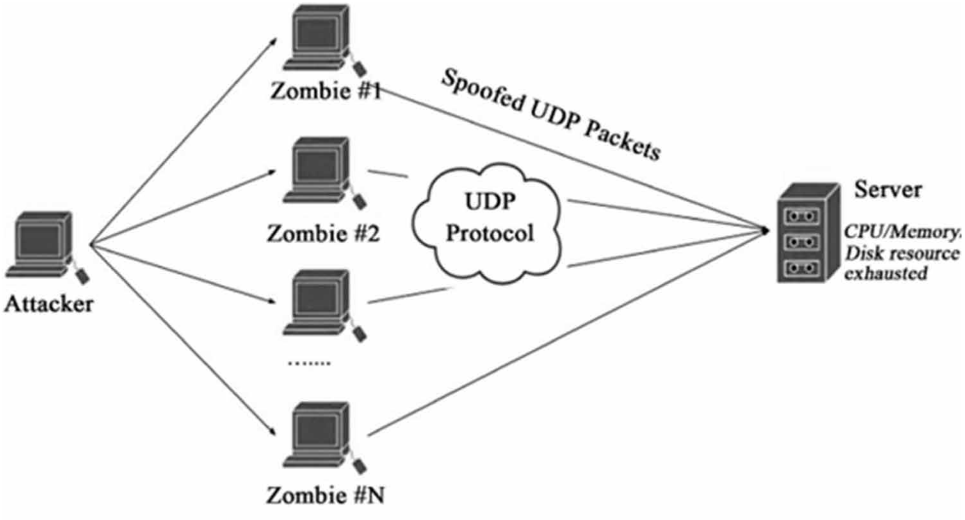
*Figure 2. Smurf attack*
*Source: Macfarlane et al. (2015)*


Logical Link
Attacker
Amplifing Network
ICMP Directed Broadcast with spoofed source IP address
ICMP requests and replies
Amplified ICMP replies
Logical Link
Target

### 2.3.3 UDP Flooding Attack

UDP flooding is a type of DDoS that makes use of the transport layer connectionless protocol UDP. A UDP flood is a denial-of-service attack in which a large number of User Datagram Protocol (UDP) packets are sent to a targeted server with the aim of overwhelming that device's ability to process and respond to the packets as shown in Figure 3. After receiving the UDP request, the server checks whether any running programs are listening for requests at the specified port. If no programs are receiving packets at that port, the server responds with an ICMP packet to inform the source that the destination is unreachable. However, if the server is flooded with UDP requests, the server becomes overwhelmed with processing and responding to UDP requests (Alzahrani, 2018). The server will fail to process legitimate requests.

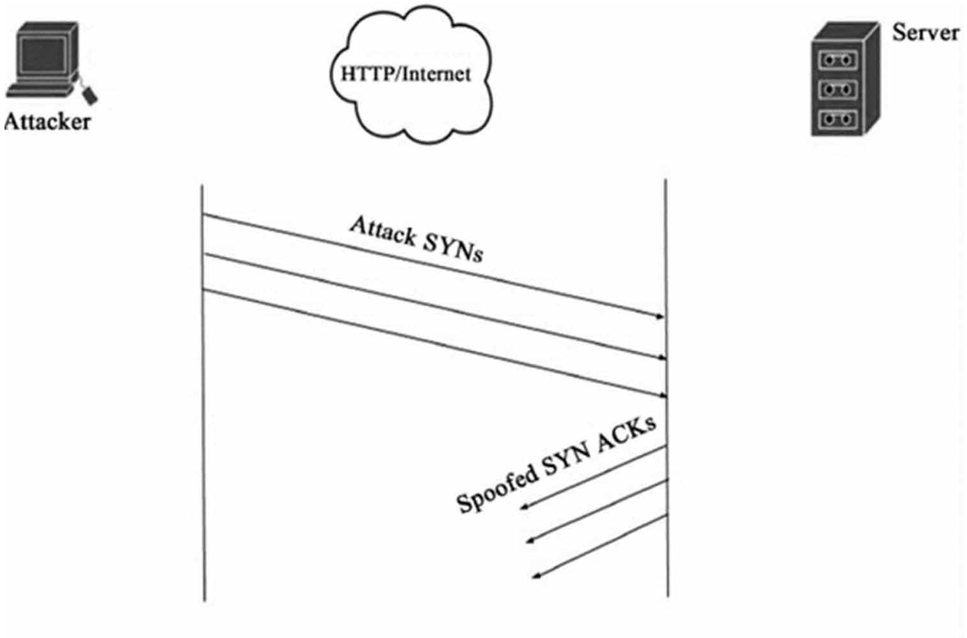
*Figure 3. UDP Flooding attack*
*Source: Alzahrani and Hong (2018)*



### 2.3.4 SYN Attack

SYN (Synchronize) attack takes advantage of how the transport layer TCP protocol works. The attack takes advantage of the TCP three-way handshake procedure designed to establish communication between hosts. SYN attack works by flooding the victim with unacknowledged SYN messages. A TCP SYN flood attack violates the operation of the TCP protocol where instead of an SYN request being answered by an SYN-ACK response, multiple SYN requests are sent to the target forcing it to wait for responses, which will never come, thereby reserving its resources until the response is received. This causes the victim machine to use up memory resources which were supposed to be allocated to genuine users. Figure 4 shows a simple diagram of a TPC SYN flood attack.

*Figure 4. TCP-SYN attack*
*Source: Alzahrani and Hong (2018)*



### 2.3.5 HTTP Flood Attack

HTTP flood attack takes advantage of the application layer protocol HTTP. The attack exploits the HTTP GET or POST requests to attack a server. The ultimate goal of this attack is to exhaust the resources of the targeted victim. The attacks target the layer where web pages are generated on the server in response to HTTP requests. A single HTTP request is easy to execute in the browser of a client but is more complicated for the target server to respond to. The server often has to load several files and may run some database queries to create a full web page. HTTP attack is analogous to repeatedly refreshing a web browser on many different computers at once which results in large numbers of HTTP requests flooding the webserver. Figure 5 shows a simplified diagram of an HTTP flood attack.

### 2.3.6 DNS DDoS attack

Domain Name Service is a UDP based system used for mapping internet domain names to internet IP addresses. A DNS DDoS amplification attack is an application layer attack which uses widely available DNS servers to amplify the attacking traffic. The attacker creates multiple DNS queries with a spoofed source IP address which are directed towards amplifying DNS servers, as shown in Figure 6. The amplifying DNS servers all direct their requests to the targeted DNS server, thereby overwhelming it so that it fails to resolve domain names for legitimate users (Macfarlane, 2015).

*Figure 5. HTTP attack*
*Source: Alzahrani and Hong (2018)*

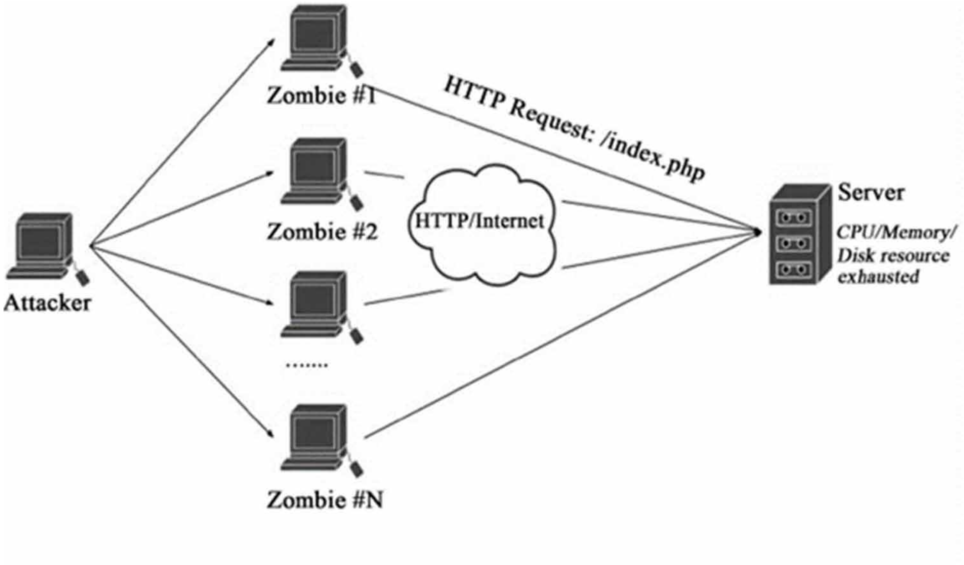
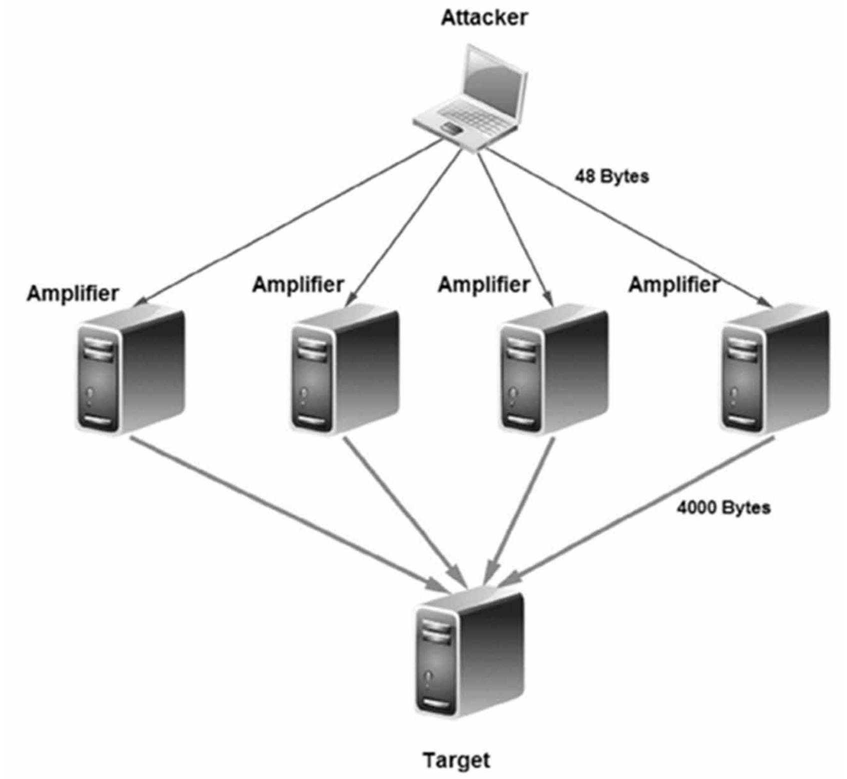


*Figure 6. Domain name service (DNS) attack*
*Source: Macfarlane et al. (2015)*



## 2.4 An Imbalanced Data Classification Algorithm of De-Noising Auto-Encoder Neural Network Based on SMOTE

Unbalanced datasets are more prevalent than balanced data sets in real-world situations. The areas in which unbalanced data sets are found include network intrusion detection, credit card fraud detection and rare disease diagnosis. In all the mentioned areas, the minority class detection rate is significant. Techniques to balance the datasets by creating synthetic samples have been studied. SMOTE is a statistical technique used to create synthetic samples for a minority class from the existing minority classes. Synthetic minority over-sampling technique (SMOTE) The SMOTE algorithm synthesizes minority class samples which do not cause over-fitting because the new samples differ slightly from the original samples. However, the samples created by SMOTE are noisy. Zhang et al. proposed a Stacked De-noising Auto-Encoder neural network (SDAE) based on SMOTE, which reduces noise through unsupervised learning (Zhang, 2016). There was a way to reduce noise in the new samples to improve classification accuracy. An Autoencoder is used by Zhang et al. to remove the noise that came with the synthetic samples. An Autoencoder is a neural network which learns about important data features and does away with irrelevant features as it represents the data in reduced dimensionality. Besides dimensionality reduction, Autoencoders are used to remove noise. The authors used Autoencoders for de-noising SMOTE generated samples. Their study is focussed on addressing the shortcomings of the SMOTE algorithm. They proposed an algorithm that de-noises the newly created SMOTE minority class samples and then trains a classifier using the balanced data. Because evaluation by accuracy gives unreliable classification performance when a classifier is trained on unbalanced data, Zhang et al. used Area Under the Curve characteristics to evaluate the performance of the SMOTE-SDAE. Experimental results show that compared with SMOTE-SVM, SMOTE-SDAE had better Area Under the Curve characteristics.

## 2.5 Intrusion Detection and Classification With Auto-Encoded Deep Neural Network

A Network Intrusion Detection System is needed to guard networks from attacks. To guard against DDoS attacks and other intrusions Rezvy et al. built an autoencoder coupled with a deep neural network (Autoencoded Deep Neural Network). Neural networks are used extensively in network intrusion detection. Neural networks can learn intricate patterns and hidden behaviours in the data. The learnt patterns and behaviours are used for differentiating benign network traffic and network attacks. A Deep Neural Network which takes input from the output of an autoencoder is used to detect network intrusions by Rezvy et al. The authors use the autoencoder for feature learning and dimensionality reduction. In their research, the autoencoder reduces data dimensions from 122 features to 61. A Deep Neural Network is then used to classify the network traffic. The data set had to be balanced using oversampling of the minority class and under-sampling of the majority class. The overall accuracy of the system for detecting network attacks was 99.3% (Rezvy, 2019).

For this research balancing the dataset is not needed because the proposed Variational Autoencoder based model is trained directly using an unbalanced network traffic data set. Accuracy is not used as a metric in this research because the data set is unbalanced, precision and recall are used instead. The ability of VAE models to learn from unbalanced data comes from the characteristic property of VAEs that they learn the underlying distribution of the data set instead of mapping encodings of the input data as is the case with plain Autoencoders. The principles and mathematics of the VAE are discussed in section 2.5.

## 2.6 Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational Auto-Encoder and Deep Neural Network

Intrusion detection systems are critical for guarding networks against attacks. The evolving nature of attacks and imbalanced data sets limit traditional machine learning from achieving high detection accuracy. Yang et al. propose an intrusion detection model that combines an improved conditional Variational AutoEncoder (ICVAE) with a deep neural network (DNN). The trained ICVAE decoder generates new minority samples to balance the training data set. The deep neural network is then trained with balanced data. The ICVAE-DNN achieved an accuracy, recall and precision of 75.43, 72.86%, and 96.20% respectively when evaluated on the NSL-KDD dataset (Yang., 2019). This research seeks to demonstrate a VAE based model that achieves higher precision and recall for detecting DDoS attacks in the unbalanced CICIDS 2017 Friday Afternoon data set.

## 2.7 Improving AdaBoost-Based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset

To improve the performance of classifiers for intrusion detection, Yulianto et al. applied the use of the Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset classes before they trained the AdaBoost based classifier. By generating more synthetic samples, the dataset becomes balanced. The results from their study show that their proposed system produces accuracy, precision, recall, and F1 score of 81.83%, 81.83%, 100%, and 90.01% respectively. (Yulianto, 2018). A recall value of 100% means their system was well-optimized against false negatives, for guarding against DDoS attacks, precisely its precision which needs to be very high. This is because, for DDoS attacks, a few false negatives do not cause harm. In this research, the aim is to achieve the highest precision so that a minimum amount of benign traffic may be misclassified. This is important where the built system is deployed in a real-world environment where mechanisms may be added to disconnect DDoS traffic automatically. While oversampling of the minority class helps to balance the classes, the newly added samples come with noise. This research chooses the Variational Autoencoder, which trains models directly on unbalanced data to achieve high precision and recall.

## 2.8 Detecting Distributed Denial of Service Attacks Using Data Mining Techniques

Alkasassbeh et al. used three machine learning algorithms to classify DDoS attacks. The authors applied Multi-Layer Perceptron, Random Forest, and Naïve Bayes to classify Smurf, UDP-Flood, HTTP-Flood and SIDDOS attacks. The accuracy of their models was 98.63%, 98.02% and 96.91% for MLP, Random Forest and Naïve Bayes, respectively. (Alkasassbeh, 2016). Although the MLP classifier achieved the highest accuracy rate this research did not choose MLP because it needs to have a balanced data set to learn from as is the case with Random Forest and Naïve Bayes. Random Forest and Naïve Bayes were not chosen because the two algorithms cannot handle large scale data sets; neither can the algorithms handle high dimensional data. This research chose direct model training without balancing the data before training. This was achieved through the use of a Variational Autoencoder. Doing away with multiple stages such as data balancing and feature engineering should result in fast training and detection should the built model be deployed in an environment in which continuous training is needed.

## 2.9 Flow Based Solutions for DoS and DDoS Attack Detection

Payload based and flow based solutions have been studied for detecting Distributed Denial-of-service attacks. Instead of using Payload based technique for identifying SYN flooding and other DDoS attacks, Noble & Sujitha find it more suitable to use Flow based detection process. In Flow based detection process the values of features are analysed instead of the Payload. The values of protocol fields such source IP, source port address, destination IP and destination port address give more information than the contents of each packet. This is especially true for DDoS because DDoS packets do not usually carry payloads that are characteristically different from those of good packets.

Noble & Sujitha proposed a flow based system that would detect three specific DDoS attacks namely Heavy Hitter (HH), Global ice berg and TCP Syn Flooding attack. In their system network traffic was grouped into flows and based on the characteristics of the flows the three DDoS types could be detected. In their implementation, an attack on an Apache web server was launched using artificially generated packets from various systems (Noble & Sujitha, 2015). In this research, a flow based approach was also taken. The Computer network traffic is grouped into flows in the CICIDS 2017 data set (Sharafaldin, 2018). The computer network flows in the CICIDS 2017 data set are used to train the VAE model in this research. A flow based detection system that can detect all types of DDoS generated from the network, transport and application layers of the TCP/IP protocol suite is adopted in this research.

## 2.10 An Adaptive Multi-Layer Botnet Detection Technique Using Machine Learning Classifiers

Malicious actors use botnets to carry out many types of intrusions including DDoS attacks.

Botnets are networks of remotely controlled compromised computers. Malicious actors can take control of them as a group and command them to carry out a specific malicious task such as launching a DDoS attack on targeted servers. Botnets are used to take advantage of Internet Relay Chat (IRC) protocol using a client-server networking model. Nowadays botnets usually use the much harder to detect Peer-to-Peer networking model. Detecting botnet traffic using Application layer protocols such as Internet Relay Chat (IRC), Hypertext Transfer Protocol (HTTP) and other types of botnets is easier because Peer-to-Peer every machine is both a server and a client to others (Khan, 2019).

. Khan, et al., propose a hybrid detection system for detecting botnet traffic. The researchers' multilayer hybrid system for Peer to Peer botnet detection combined multilayer perceptron and Decision Tree algorithm. Session features were used to classify the benign P2P communication and P2P botnet communication. They first filtered out non-peer-to peer traffic in the first layer of the hybrid system. At the second layer, they further characterized peer-to-peer and non-peer-to-peer network traffic. At the third layer the performed dimensionality reduction to leave only features which explain botnet Peer-to-Peer traffic. At the final layer, they performed classification using Decision Trees. Their hybrid system was based on both traffic behaviour and network flows similarity. Validated on CTU-13 and ISOT datasets the researchers' model was able to detect botnet traffic with an accuracy of 98.7% (Khan, 2019). Although it was observed that the Decision Tree algorithm had a high accuracy to detect Peer-to-Peer botnet traffic they cannot handle high-dimensional data sets, they are not an option for this research. Because of the increase of connected devices from the Internet of Things and the resultant increased network traffic, it is essential that detection models can handle large numbers of tuples of big data scales. Decision tree models cannot handle large scale data sets. For the above two limitations, VAE is chosen ahead of decision trees for this research.

## 2.11 Best Features Based Intrusion Detection System by RBM Model for Detecting DDoS in Cloud Environment

Most Intrusion Detection Systems (IDS) are computationally demanding and have poor detection accuracy. To detect DDoS attacks more accurately Mayuranathan et al built a Deep learning Restricted Boltzmann Machines (RBM) system which is not computationally demanding by using only the best feature set in the data set. Their system uses a Random Harmony Search (RHS) optimization model to identify the best set of features needed to detect DDoS attacks. Using the reduced dimensions of only the best set of features they build a Deep learning classifier to detect DDoS attacks. (Mayuranathan, 2019). The RHS-RBM system achieves an accuracy of 99.96%. This research proposes a DDoS detection without the prior stage of data pre-processing to select features.

## 2.12 Variational Data Generative Model for Intrusion Detection

Intrusion Detection systems need balanced and diversified data to train valid classifiers. An Intrusion detection which can generate new synthetic samples to balance the data set by using a Conditional Variational Autoencoder is proposed by (Lopez-Martin, 2018). The synthesized samples are better than SMOTE generated samples because they have a probabilistic similarity to the original samples (Lopez-Martin, 2018). Their study was focussed on detecting network intrusions in the Internet of Things domain. The authors' method is based on a conditional variational autoencoder with a decoder designed to generate new samples according to specified labels. The generative VAE uses the reconstruction error of the decoder as a measure to determine the class of network traffic. The approach is anomaly-based supervised machine learning. Labels that were below a particular reconstruction error were classified as intrusion. Their system is a model taking in features and a class label as input (Lopez-Martin, 2018). A plain VAE takes in only features as input and learns the distribution of the features. Lopez-Martin et al added a label as input so that the CVAE could be conditioned to generate a specified label. Evaluation of the classification results showed that their intrusion detection system had an accuracy of 80.10%. This research will train a DDoS detector using a plain VAE which takes in features only as input.

## 2.13 Generative Models

Machine learning motivation can be classified into discriminative and generative. Discriminative machine learning aims to learn a function that, given the input feature values, would predict the class of the input tuple. With generative modelling, the aim is to learn the joint underlying distribution of all variables. Generative models aim to understand how the data was created. Understanding the data generating process makes classification easier for a classifier that is coupled to a generative model (Kingma, 2019). Generative models are instrumental in dealing with challenges which have traditionally beset intrusion detection such as data set imbalances and lack of intrusion samples (Kingma, 2019) There are three types of deep learning generative models namely Autoencoders, Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN). An Autoencoder is a type of neural network that is trained to reproduce its input at the output layer. The number of neurons in each hidden layer progressively gets fewer thereby representing the input by fewer features at each layer. The autoencoder aims to represent the input in a lower dimensional space before trying to reconstruct the same input at the output. If it succeeds in reconstructing the input at the output using lower dimensions it means the rest of the discarded features or dimensions were redundant or noise. The probability of reconstructing an input accurately can be used to classify network traffic in intrusion detection. This means an Autoencoder learns the most important features that describe classes in a dataset. Besides dimensionality reduction, Autoencoders are also used for denoising data. A Variational autoencoder takes in samples and represents the data as a continuous joint probability distribution in the latent space. During training the joint probability distribution is forced to follow a normal distribution by minimizing a KL divergence loss. Classes within the data set occupy different regions of the continuous probability distribution (Kingma, 2019) . The probability distribution of a VAE can be sampled to generate new data points as is the case with conditional variational autoencoders. Generative Adversarial Networks (GANs) are a third type of generative models that map an input to a lower dimensional latent space. The data distribution in a GAN is not continuous. Therefore a GAN cannot create stable and new samples from its distribution. A discriminator forces the output of a GAN to be as close as possible to the input. This paper classifies DDoS attacks from benign by applying both linear and quadratic decision boundaries to the latent representation of a VAE.

## 2.14 Variational Autoencoder

VAEs are built on top of deep neural networks. Like all deep neural networks, they are standard function approximators, which are trained fast with mini-batch stochastic gradient descent. No strong assumptions are required for implementing a VAE model. The main assumption that features are normally distributed is usually the case in many real world data sets and in models that use large scale data sets (Doersch, 2016).

Complicated dependencies among dimensions make training of classification models difficult. VAE is based on the assumption that multidimensional data can be projected onto a probabilistic lower dimensional subspace. In the lower probabilistic representation of the input, benign and DDoS attacks are more easily separable (Ikeda, 2018). In this research a decision boundary is fit to the latent space representation of the network flows to separate DDoS and benign attacks.

VAEs have also found extensive use in image and audio generation. They are particularly important because of their ability to generate new samples which are similar but not the same as any in the original dataset. VAEs create new audio and image files which share the same data distribution as the original dataset. VAEs are also as novel way of classifying imbalanced data with high precision and high recall. The ability of VAE to classify will be demonstrated in this research. Several techniques have been developed to improve the performance of Machine Learning classification algorithms when learning from class imbalanced data. Machine Learning algorithms for intrusion detection usually have very few anomalous examples to learn from. The model therefore learns more from the majority of benign samples. This negatively affects the predictive performance of the classifier in detecting intrusions. The most used methods of addressing class imbalances are sampling techniques (Fajardo, 2018). Oversampling of the minority class results in overfitting whereas under sampling of the majority class results in the loss of vital information needed for training the model (Jiadong & Wang, 2019). SMOTE is an improvement in creating synthetic samples that enhance model learning. However, SMOTE samples produce noisy samples as described in section 2.1. A VAE offers a deep learning approach that does not require the data set to be balanced before training the classifier. VAEs can classify imbalanced data with high precision and high recall i.e. both false positives and false negatives are minimized respectively. In this research, the VAE network learned how to represent the input data as a Gaussian distribution. Linear and Quadratic Discriminant Analysis were applied to the re-parameterized values of the data points. Like the VAE algorithm, the LDA and QDA algorithm also assumes that the dataset features have a normal distribution (Ghojogh & Crowley, 2019). VAEs are especially useful in that they can separate and classify data points in situations where we have unlabelled data and imbalanced data. All traditional Machine Learning algorithms (Support Vector Machines, Decision Trees and Logistic Regression) find it difficult to train classifier imbalanced data (Yilmaz, 2019). VAEs are highly applicable to both Supervised and Unsupervised learning. In unsupervised Deep Learning VAEs take an anomaly detection approach and are trained using only benign data samples. After Learning the data distribution of only the benign samples the trained VAE has low reconstruction probabilities for data points with distributions that do not conform to its learnt distribution. Data points below a set probability threshold are classified as anomalous. A data point with a similar distribution to the one learnt by the VAE has a high reconstruction probability and is classified as benign (An & Cho, 2015). VAE networks can also be trained in a Supervised Machine Learning way and be conditioned to generate new synthetic samples of the required class. This research did not use VAE to generate new samples but only used VAE to classify DDoS traffic by applying LDA and QDA to the latent representation. The VAE is especially suited to DDOS detection because of its scalability. VAE can handle Big Data scale datasets which are more likely to be encountered in computer networks. The scalability of VAEs is enabled by the re-parameterization trick which makes it possible to update weights during training. The applicability of Stochastic Gradient Descent in updating weights make VAE scalable to any dataset size. VAEs apply to all anomaly detection scenarios from network intrusion detection to credit Card Fraud detection and rare disease diagnosis. Kurien and Chikkamannur applied VAE to detect real time anomalies in credit card fraud detection. The results from their study showed that the VAE model they built performed better than Logistic Regression and Random Forest ensemble models (Kurien, 2019). In this paper, VAE is applied to detect network intrusion (DDoS attacks) in computer networks. Jorge et al. showed that data augmentation using Variational Autoencoders resulted in models which were generalizable. They also demonstrated that classification accuracy improved when synthetic data points were generated by VAE (Jorge, 2018).

A variational autoencoder (VAE) is a generative combination of two neural networks. The first neural network is the encoder which takes in data inputs and represent the data in a latent space with reduced dimensions. The latent space is in the form of a joint probability distribution of the important variables. The distribution is forced to be normal by minimization of a KL divergence term. The second neural network takes samples from the latent space as input and attempts to reconstruct the original input. The difference or dissimilarity between the reconstructed output and the original input is a loss to be minimized. During training the loss function of the VAE minimizes the sum of the KL divergence loss and the reconstruction loss. The latent space variables are used by the decoder of the VAE to generate data samples similar to the input data. The latent space representation is a joint probability distribution of all the features. Through the use of the re-parameterization trick (explained in Section 2.7) a VAE model updates latent variables using stochastic gradient descent. Because a VAE models the underlying data distribution of the classes found in the dataset, VAEs can learn to separate the classes. New samples can be generated by sampling from the vector of means and vector of variances in the latent representation. The sampled mean and variance are passed through the decoder to generate new samples (Kurien, 2019).

In this research, we used the decoder part of the VAE only during training. When the model was trained we applied decision boundaries to the variables of the latent space.

Figure 1 shows the structure of a VAE encoder. In a VAE, the highest layer of the model $\varepsilon$ is treated as the latent variable where the generative process starts. *g(z)* represents the process of data generation that results in the data x. The marginal likelihood of a data point is the sum over the marginal likelihood of individual data points

$$\log p_\theta\left(x^{(1)}\right),\ldots..,x^{(N)}) = \sum_{i=1}^{N} \log p_\theta x^{(i)} \tag{1}$$

where the marginal likelihood of individual data points can be rewritten as

$$\log p_{\theta(x)^{(i)}=} D_{KL}(q_\theta\left(z|x\right) \,||\, p_\theta\left(z\right) + \mathcal{L}\left(\theta,\phi;x^{(i)}\right) \tag{2}$$

Where $q_\phi(z|x)$ is the approximate posterior and $p_\theta(z)$ is the prior distribution of the latent variable $z$. Since marginal likelihood is intractable, the variational lower bound of the marginal likelihood of data is taken as the objective function of a VAE. $\mathcal{L}\left(\theta,\phi;x^{(i)}\right)$ is the variational lower bound on the marginal likelihood of the data point i. The variational lower bound allows joint optimization concerning $\theta$ and $\phi$ using mini-batch stochastic gradient descent (SGD). Starting with random initial values of $\theta$ and $\phi$, their values are stochastically optimized until convergence (Kingma, 2019). DKL($q_\theta(z|x)||p_\theta(z)$ is the KL divergence of the approximate posterior and the prior.

Since the KL divergence term is at least 0, the equation can be rewritten as follows

$$\log p_\theta(x^{(i)} \geq \mathcal{L}(\theta,\phi;x^{(i)} \tag{3}$$

$$= E_{q\phi}\left(z|x^{i}\right)\left[-\log q_\phi\left(z|x\right) + \log p_\theta\left(x|z\right)\right] \tag{4}$$

$$= -D_{KL}(q_\phi\left(z|x^{i}\right) \,||\, p_\theta(z) + E_{q\phi}\left(z|x^{i}\right)\left[\log p_\theta\left(x|z\right)\right] \tag{5}$$

$p_\theta(x|z)$ is the likelihood of the data x given the latent variable z$= -D_{KL}(q_\phi\left(z|x^{i}\right) \,||\, p_\theta(z$ is the KL divergence between the approximate posterior and the prior of the latent variable $z$.

The KL divergence term forces the posterior distribution to have a normal distribution. In this way, the KL divergence works as a regularization term. $E_{q\phi}\left(z|x^{i}\right)[\log p_\theta\left(x|z\right)$ represents the reconstruction of $x$ through the posterior distribution $q_\phi(zx)$ and the likelihood $p_\theta(x|z)$. Parameters $\phi$ and $\theta$ are updated during training for the encoder and decoder networks respectively. When the loss function is at its lowest, the classes occupy distinct but continuous regions of the probability distribution. In this research near and quadratic boundaries are applied.

*Figure 7. Encoder of VAE*
*Source: An and Cho (2015)*

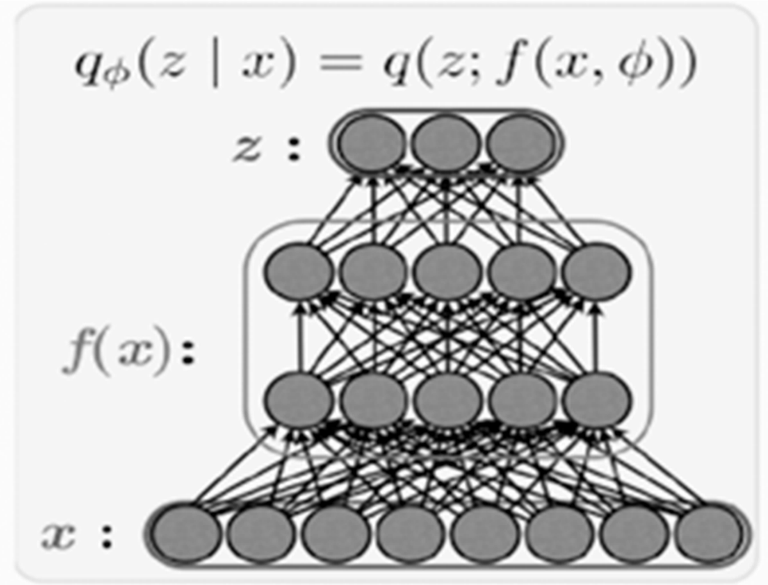
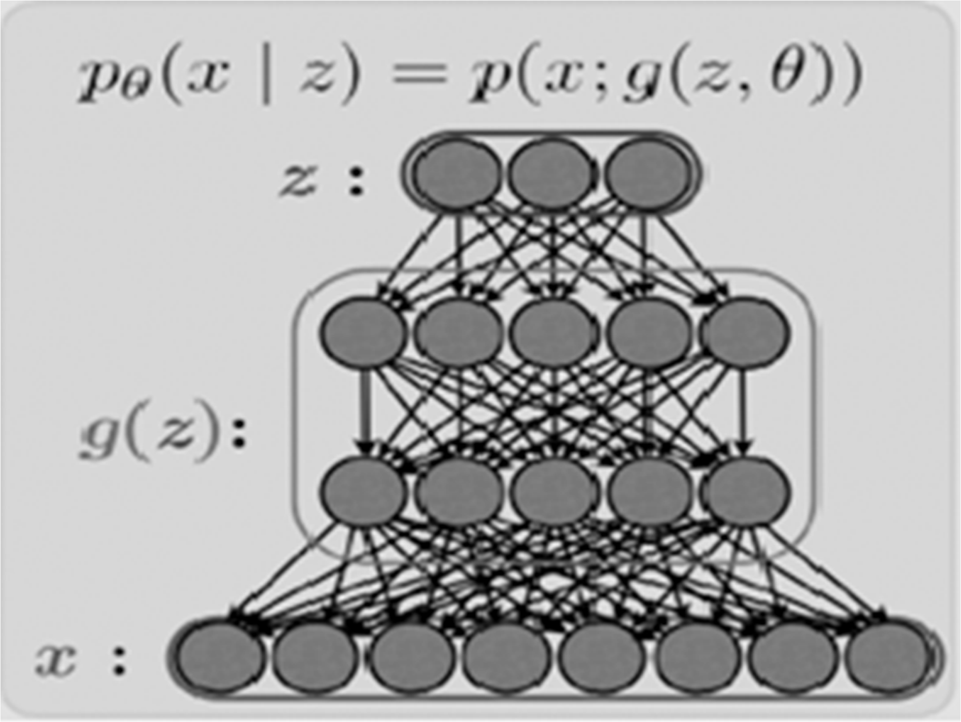


*Figure 8. Decoder of VAE*
*Source: An and Cho (2015)*



## SNU Data Mining Center

The approximate posterior $q_\phi(z|x)$ represents the encoder whereas $p_\theta(x|z)$ represents the decoder. The decoder is a directed probabilistic graphical model. VAE models the parameters of the data distribution, not the value itself. $f(x,\phi)$ in the encoder outputs the parameters of the approximate posterior $q_\phi(z|x)$. To get the actual value of the latent variable $z$, samples are taken from $q(z; f(z,\phi))$. The encoder and decoder are probabilistic. $f(x,\phi)$ being a neural network represents the transformation from the data x to the latent variable $z$. To get the reconstruction $\hat{x}$, given the sample $z$, the parameter of $p_\theta(x|z)$ is obtained by $g(z,\theta)$ where the reconstruction $\hat{x}$ are sampled from $p_\theta(x; g(z,\theta)$.

This research will highlight three ways in which a VAE can be very useful in DDoS intrusion detection. VAE can be useful in situations where we have unbalanced classes in data sets. In this case, a VAE enhances the separation of classes when they are represented in lower dimensional joint probability distribution. This is the approach that was used in this research. A VAE can also be useful in unsupervised machine learning anomaly detection where there is a total lack of malicious samples to learn from. Only benign network flows are used to train the model that would detect anomalies. In this application, a data point is defined as an anomaly or an intrusion if the probability of it being generated from the model is below a certain threshold. The model outputs probability as the decision rule for judging anomalies. The trained model detects anomalies by their deviation from the learnt probability model (Wanga, et al., 2019). The third way a VAE can be used to deal with class imbalance is by generating minority samples to balance the data for downstream use by traditional classifiers.

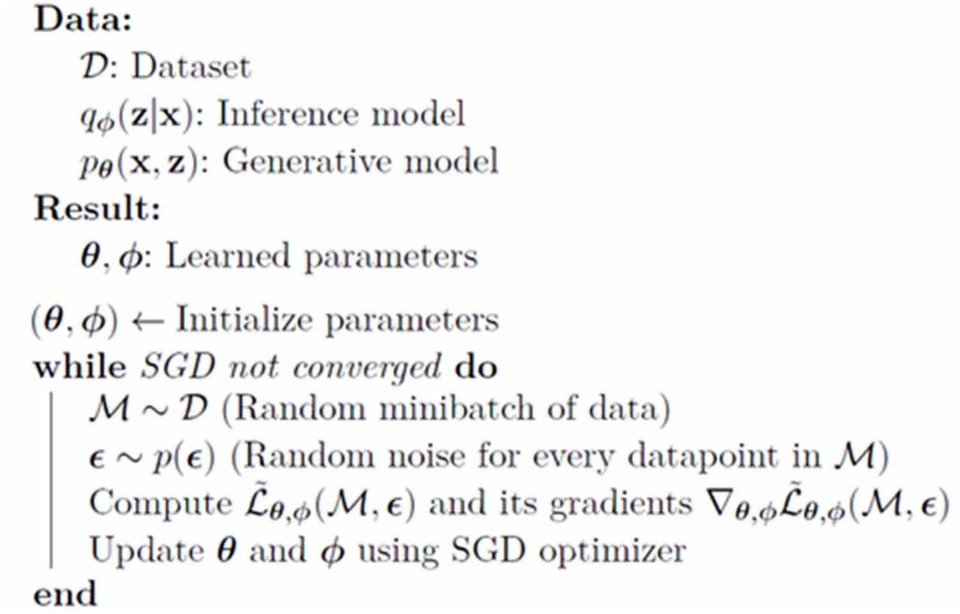
## 2.15 Re-Parameterization Trick

Backpropagation cannot pass through random variables of the latent space because the latent space is a probability distribution that is not differentiable. The re-parameterization trick is a way to make the representation of the latent space differentiable. Making the latent space representation differentiable and deterministic makes it possible to update the value of $\phi$ for every mini batch of data samples fed into the encoder. Re-parameterization removes the randomness from the latent space and externally applies it as epsilon as shown in Figure 10. With re-parameterization, a sound mathematical way was found to go round the randomness of the VAE latent space z in updating $\phi$ parameters of the encoder during training. To re-parameterize the random variable $z \sim q_\phi(z|x)$ was transformed into a differentiable transformation of another random variable given $z$ and $\phi$:

$$z = g(\varepsilon, \phi, x)$$

$\varepsilon$ is *a* random variable whose distribution is independent of $x$ or $\phi$. The algorithm is summarized as:

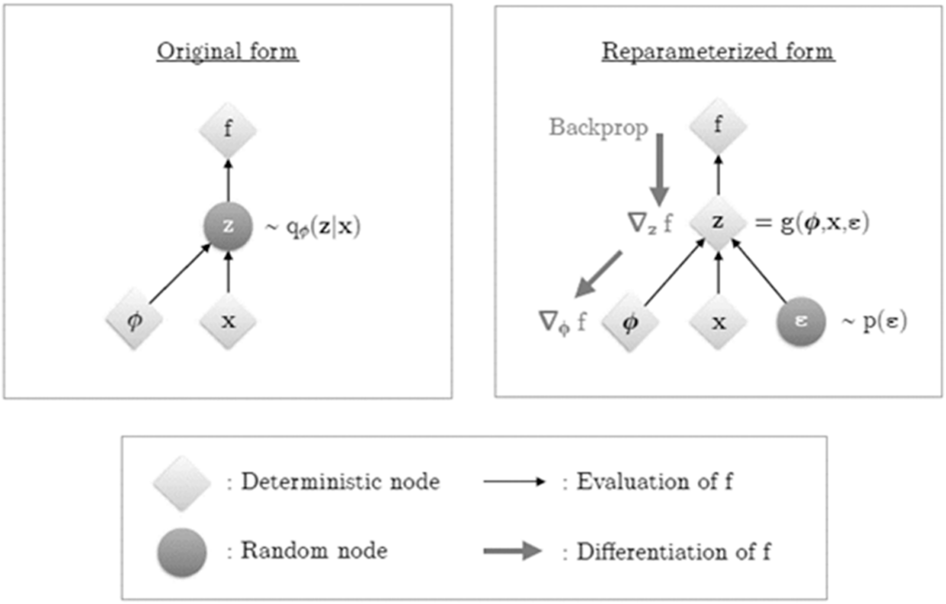
*Figure 9. VAE re-parameterization algorithm*
*Source: Kingma and Welling (2019)*



```
Data:
    $\mathcal{D}$: Dataset
    $q_{\phi}(\mathbf{z}|\mathbf{x})$: Inference model
    $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$: Generative model
Result:
    $\boldsymbol{\theta}, \boldsymbol{\phi}$: Learned parameters

$(\boldsymbol{\theta}, \boldsymbol{\phi}) \leftarrow$ Initialize parameters
while SGD not converged do
    $\mathcal{M} \sim \mathcal{D}$ (Random minibatch of data)
    $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ (Random noise for every datapoint in $\mathcal{M}$)
    Compute $\tilde{\mathcal{L}}_{\boldsymbol{\theta},\phi}(\mathcal{M}, \boldsymbol{\epsilon})$ and its gradients $\nabla_{\boldsymbol{\theta},\phi}\tilde{\mathcal{L}}_{\boldsymbol{\theta},\phi}(\mathcal{M}, \boldsymbol{\epsilon})$
    Update $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using SGD optimizer
end
```

*Figure 10. Illustration of the re-parameterization trick*
*Source: Kingma and Welling (2019)*



### 2.16 Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA)

LDA aims to discriminate different classes by projecting the data on to a low dimensional space, retaining only the feature values that enhance the separation of the classes found within the data. LDA is a probabilistic statistical supervised binary classification method. The analysis technique applies a linear decision boundary between classes. LDA assumes that the two classes have a normal distribution. The assumption is not restrictive because normal distribution is the most common distribution found in real-world data (Ghojogh & Crowley, 2019). The classification technique was first proposed by Ronald Fisher in 1936. The technique seeks to maximize the function that represents the difference between the means of the two classes. The function is then normalized by a measure of the within-class variability. He correctly reasoned that maximising the distance between the means of each class and minimising the spreading within the individual classes would separate the classes.

The distance between the means of the two classes is calculated first. The distance is called the between-class variance. The distance between the mean and the samples of each class is then calculated. This distance is called the within-class variance. A new dimensional space which maximizes the between-class variance and minimizes the within class variance is sought. LDA projects the original data onto a new dimensional space where class separation is enhanced. The classes are separated in the projected new feature space (Tharwat, 2017). The two main assumptions of LDA are that the features are normally distributed and that the covariance matrix of the two classes is equal. QDA also assumes that each class is drawn from a normal multivariate distribution. The difference with LDA is that QDA assumes that the two classes have two different covariance matrixes. When these assumptions are satisfied QDA approximates the Bayes classifier accurately and the discriminant function produces a quadratic decision boundary. QDA classifies better than LDA if the covariance matrix is not the same in the two classes. For large datasets, the assumptions of both QDA and LDA can be violated and still achieve high classification performance. Quadratic Discriminant Analysis. In this research, LDA and QDA are applied to the latent representation of the data to separate and classify the dataset into benign and DDOS attacks.

## 3, RESEARCH METHODOLOGY

### 3.1 Overview of the Method of Investigation

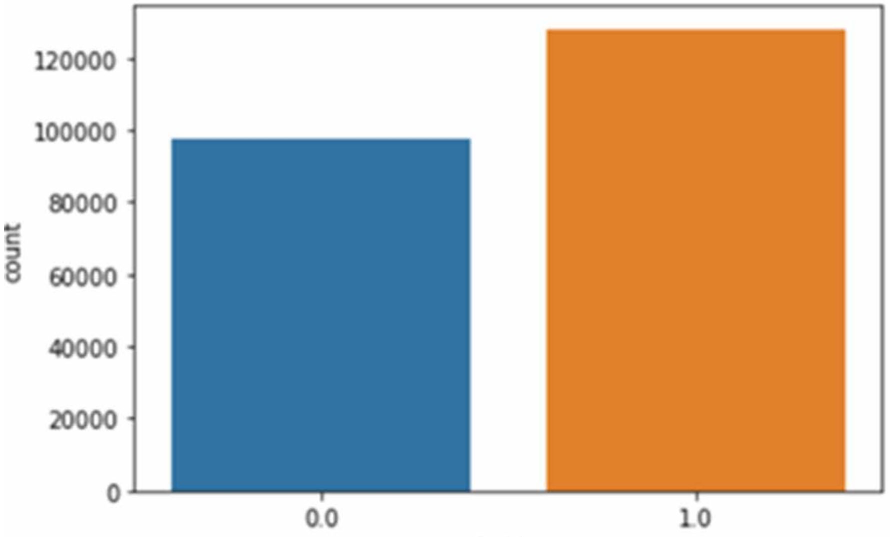
A VAE is trained with computer network flows containing DDoS and benign traffic flows. The trained VAE takes the input of 77 features of the CICIDS Friday Afternoon data set. The VAE learns the underlying distribution of the data and encodes each network flow into a mean and standard deviation. The mean and standard deviation represent the position of the network flow in the joint probability distribution. Quadratic Discriminant Analysis, Linear Discriminant Analysis and Deep Neural Network then take input of means and standard deviations and train classifiers. QDA applies a quadratic decision boundary, and LDA applies a linear decision boundary to separate the benign and DDoS flows. It is assumed that the model with high precision and recall between QDA and LDA fulfils its attendant assumptions more than the one with lower precision and recall. The assumptions that go with QDA and LDA will be brought into focus when comparing the classification performance of VAE based QDA and LDA Section 4 Results and Discussion. The classification performance of the Discriminant Analysis with higher precision and recall is further compared to the classification performance of the Deep Neural Network. A Hybrid DDoS classifier is being built using the foundation of the VAE. The final model includes both a Discriminant Analysis classifier and a Deep Neural Network. The two classifiers serve to endorse each other's results.

## 4. RESULTS AND DISCUSSION

### 4.1 Introduction

A confusion matrix was used to determine the precision and recall of the built models. The classification performance of the built VAE based Deep Neural Network, Linear Discriminant Analysis and Quadratic Discriminant Analysis classifiers were evaluated on the test data using a confusion matrix. A confusion matrix was shown for each evaluation of the built model. Linear Discriminant Analysis, Quadratic Discriminant Analysis and Deep Neural Network algorithms classifiers were trained on the VAE latent representation of computer network traffic flows. Different hyper parameter settings were used to come up with the best model for the VAE and the Deep Neural Network. Different hyper parameters were tried in different combinations until the best performing model was achieved. The precision and recall of the three classifiers were compared against each other and also against the results Sharafaldin et al. got from traditional machine learning classification algorithms as recorded in Table 5. The CICIDS 2017 Friday afternoon data set had a total of 225745 samples of which 97718 were benign and 12827 were DDoS attacks. The class distribution is shown in Figure 11. Linear Discriminant Analysis, Quadratic Discriminant Analysis, and Deep Neural Network were trained on network flow data that had been encoded by the VAE into means and standard deviation.

*Figure 11. Class distributions of CICIDS 2017 Friday afternoon dataset*



## 4.2. Results

### 4.2.1 Comparing VAE-Based LDA and QDA Results Against Traditional Ml Classifiers

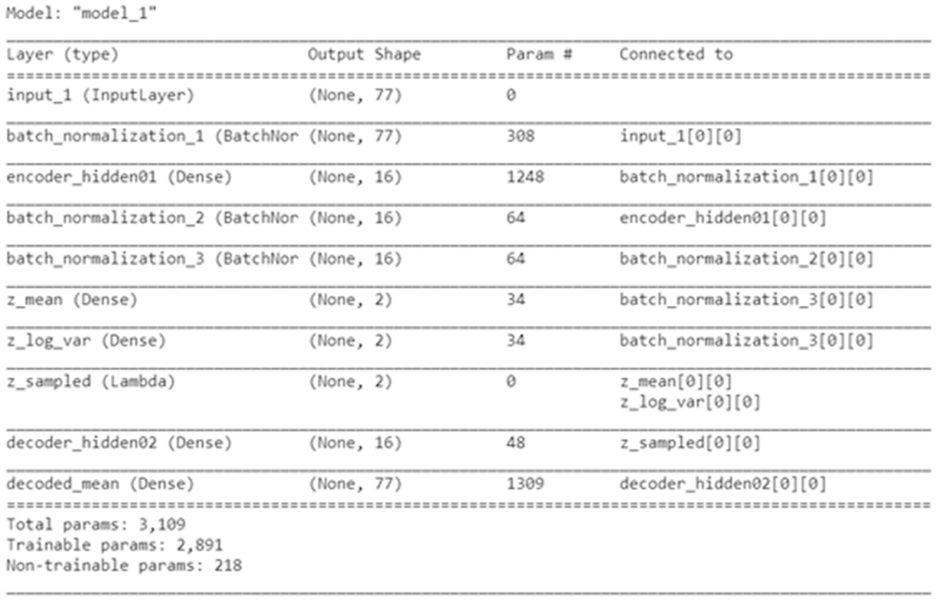
VAE with 16 neurons in encoder and decoder hidden layers.

A Linear Discriminant and Quadratic Discriminant Analysis classifier was trained on the latent representation of network flows. The VAE model was built with a single hidden layer for the encoder and decoder. In the hidden layer, 16 neurons were used. The model was trained for 100 epochs. The early Stopping feature of Keras was configured with a 'patience' value of 5, meaning the training would stop if the loss value did not decrease for five successive epochs. The summary of the built VAE model is shown in Figure 15. The distribution of the benign and DDoS attack samples in the CICIDS 2017 Friday Afternoon data set is shown in Figure 11. How the trained VAE represented the classes in the latent space was visualized in Figure 14. The obtained precision and recall of the VAE-LDA and VAE-QDA are shown in Table 3 and Table 4 respectively.

The visualisation showed that DDoS attack traffic flows (orange colour) tended to be more clustered than benign traffic flows.

The precision, and recall of the VAE-LDA are 92.2% and 99.92% respectively. A linear decision boundary resulted in 6953 false positives whereas a quadratic decision boundary resulted in 8360 false positives. Table 5 shows the classification results of traditional machine learning algorithms on the CICIDS 2017 dataset obtained by Sharafaldin et al. Sharafaldin et al. used feature engineering to select only determining features for detecting DDoS attacks. For detecting DDoS attacks the researchers used Average Packet Size, Backward Packet Length Standard Deviation, Flow Duration, Flow Inter Flow Arrival Time (IAT) and Standard Deviation as the main features with parameters whose value determined whether a network flow was benign or DDoS attack.

*Figure 12. Summary of the VAE model*



```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape          Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 77)            0
__________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 77)            308         input_1[0][0]
__________________________________________________________________________________________________
encoder_hidden01 (Dense)        (None, 16)            1248        batch_normalization_1[0][0]
__________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 16)            64          encoder_hidden01[0][0]
__________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, 16)            64          batch_normalization_2[0][0]
__________________________________________________________________________________________________
z_mean (Dense)                  (None, 2)             34          batch_normalization_3[0][0]
__________________________________________________________________________________________________
z_log_var (Dense)               (None, 2)             34          batch_normalization_3[0][0]
__________________________________________________________________________________________________
z_sampled (Lambda)              (None, 2)             0           z_mean[0][0]
                                                                  z_log_var[0][0]
__________________________________________________________________________________________________
decoder_hidden02 (Dense)        (None, 16)            48          z_sampled[0][0]
__________________________________________________________________________________________________
decoded_mean (Dense)            (None, 77)            1309        decoder_hidden02[0][0]
==================================================================================================
Total params: 3,109
Trainable params: 2,891
Non-trainable params: 218
__________________________________________________________________________________________________
```

*Figure 13. Training loss compared with validation loss*

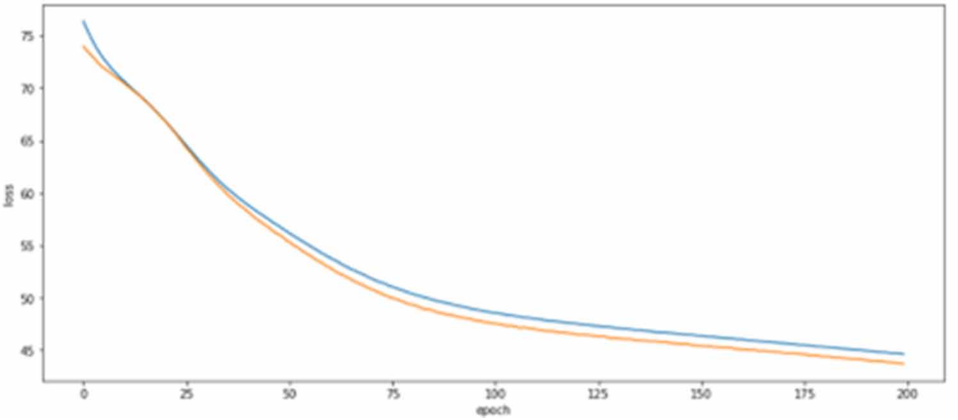


*Figure 14. Latent space visualization of training data (VAE with a single 16-neuron hidden layer)*

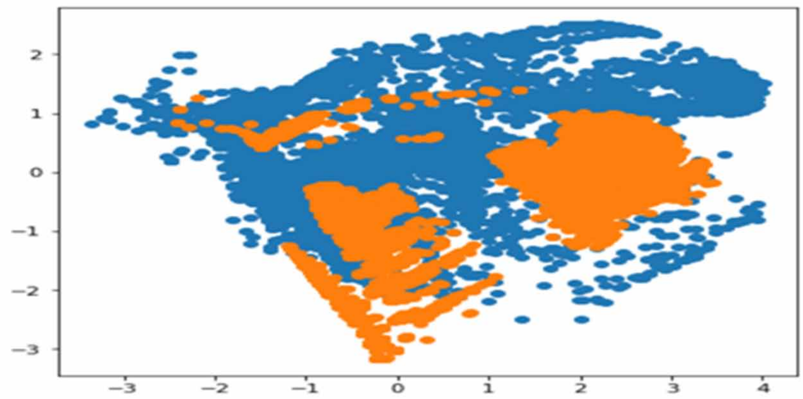


*Table 3. Classification results of LDA based on VAE model with single 16-neuron layer*

| LDA Classification Confusion Matrix Results | |
|---|---|
| AUC(ROC) | 0.924185472 |
| Precision | 0.920261933 |
| Recall | 0.999227963 |
| F1 score | 0.958120653 |
| False positives | 6953 |
| True positives | 80245 |
| False negatives | 62 |
| True negatives | 39137 |

*Table 4. Classification results of QDA based on VAE model with single 16-neuron layer*

| QDA Classification Confusion Matrix Results | |
|---|---|
| AUC(ROC) | 0.908940536 |
| Precision | 0.90565186 |
| Recall | 0.999265319 |
| F1 score | 0.950158364 |
| False positives | 8360 |
| True positives | 80248 |
| False negatives | 59 |
| True negatives | 37730 |

*Table 5. The classification performance of traditional machine learning classifiers*

| Algorithm | Precision | Recall |
|---|---|---|
| KNN | 0.96 | 0.96 |
| RF | 0.98 | 0.97 |
| ID3 | 0.98 | 0.98 |
| Ada boost | 0.77 | 0.84 |
| MLP | 0.77 | 0.83 |
| Naive-Bayes | 0.88 | 0.04 |
| QDA | 0.97 | 0.88 |

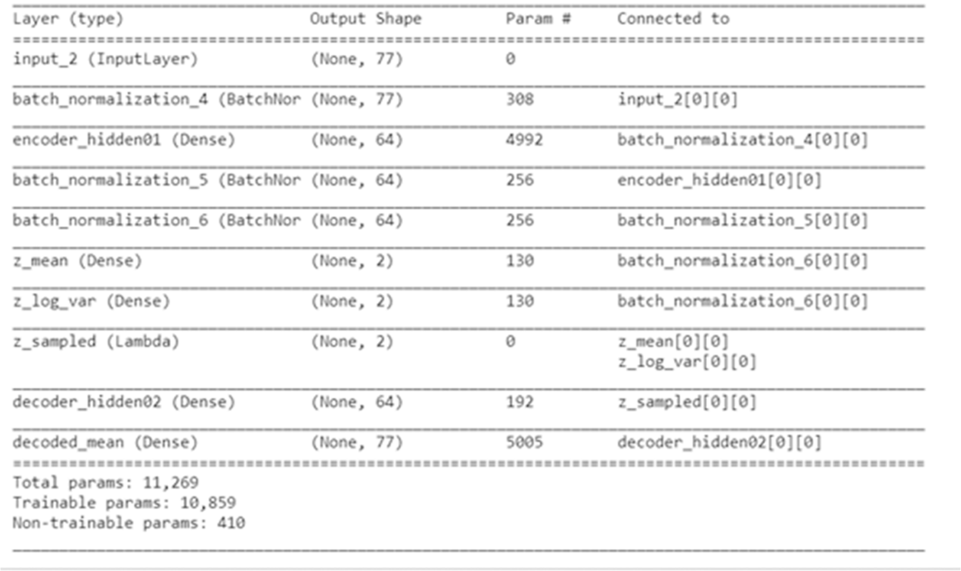
Source: Sharafaldin (2018)

The recall of VAE-LDA and the VAE-QDA was 99.92% and 99.93% respectively. The recall is higher than the recall achieved by traditional machine learning algorithms. KNN, ID3 and RF had a precision of 96%, 98% and 98% respectively compared to the VAE-LDA and VAE-QDA of 92% and 90% respectively. Random Forest had a high precision of 98%. However, because RF cannot work in high-dimensional data because the depth of the trees becomes too long. The scalability and ability to handle high dimensional data gave VAE based classifiers an advantage over traditional machine learning classifiers.

## VAE With 64 Neurons in Encoder and Decoder Hidden Layers

A second VAE-DA was built using 64 neurons in the hidden layer of the encoder and decoder layers. A batch size of 5000 was used. The summary of the built VDE-DA is shown in Figure 15 the loss obtained during training is shown in Figure 16 and the visualization of training data is shown in Figure 17.

*Figure 15. Summary of 64-neuron single hidden layer VAE model*



```
________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
================================================================================================
input_2 (InputLayer)            (None, 77)           0
________________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, 77)           308         input_2[0][0]
________________________________________________________________________________________________
encoder_hidden01 (Dense)        (None, 64)           4992        batch_normalization_4[0][0]
________________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, 64)           256         encoder_hidden01[0][0]
________________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, 64)           256         batch_normalization_5[0][0]
________________________________________________________________________________________________
z_mean (Dense)                  (None, 2)            130         batch_normalization_6[0][0]
________________________________________________________________________________________________
z_log_var (Dense)               (None, 2)            130         batch_normalization_6[0][0]
________________________________________________________________________________________________
z_sampled (Lambda)              (None, 2)            0           z_mean[0][0]
                                                                 z_log_var[0][0]
________________________________________________________________________________________________
decoder_hidden02 (Dense)        (None, 64)           192         z_sampled[0][0]
________________________________________________________________________________________________
decoded_mean (Dense)            (None, 77)           5005        decoder_hidden02[0][0]
================================================================================================
Total params: 11,269
Trainable params: 10,859
Non-trainable params: 410
________________________________________________________________________________________________
```

*Figure 16. Loss for a model with a single 64-neuron hidden layer VAE model*

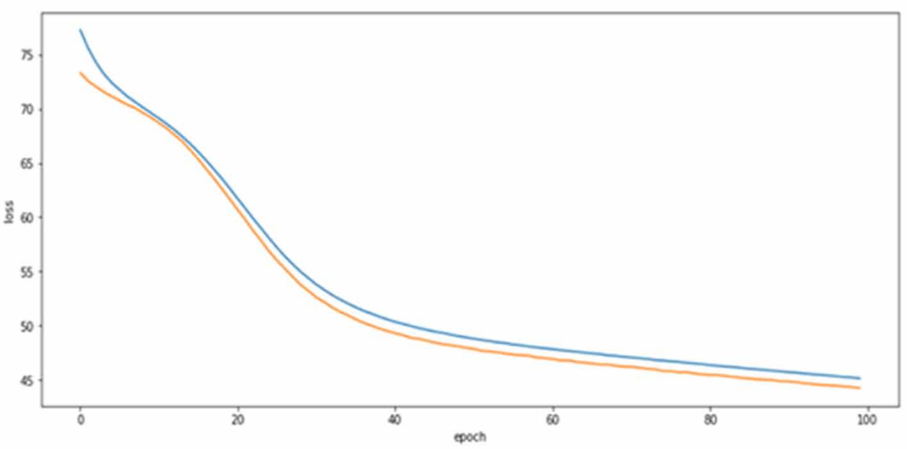


*Figure 17. Latent space visualization of VAE with a single 64-neuron hidden layer*

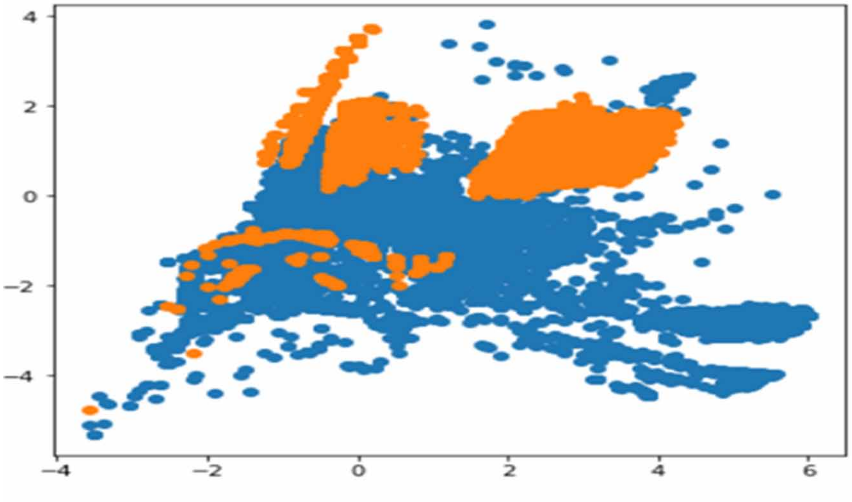


*Table 6. Confusion matrix results for single 64-neuron hidden layer VAE-LDA model*

| LDA Classification Confusion Matrix Results | |
|---|---|
| AUC(ROC) | 0.833973 |
| Precision | 0.8599 |
| Recall | 0.932733 |
| F1 score | 0.894837 |
| False positives | 12204 |
| True positives | 74905 |
| False negatives | 5402 |
| True negatives | 33886 |

*Table 7. Confusion matrix results for VAE-QDA model with single 64 neuron hidden layer for decoder and encoder*

| QDA Classification Confusion Matrix Results | |
|---|---|
| AUC(ROC) | 0.946285 |
| Precision | 0.96047 |
| Recall | 0.961523 |
| F1 score | 0.960996 |
| False positives | 3178 |
| True positives | 77217 |
| False negatives | 3090 |
| True negatives | 42912 |

For a model built with 64 neurons per hidden layer, the VAE based Quadratic Discriminant Analysis showed a precision of 96.04% and recall of 96.15% respectively. The VAE based Linear Discriminant Analysis showed precision and recall of 85.99% and 93.27% respectively. Applying a quadratic decision boundary had higher precision and recall for a model with 64 neurons trained on a batch size of 5000.

A second model was built again using 64 neurons in the hidden layers of the encoder and decoder but using a batch size of 1500. The model had the following classification performance as shown in Table 7.

*Figure 18. Loss for a model with a single 64-neuron hidden layer VAE model (batch size = 1500)*

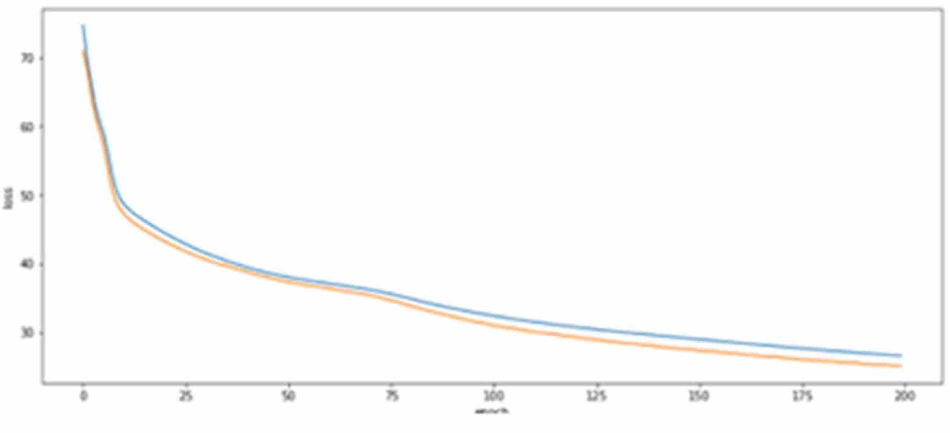


*Figure 19. Latent space visualization of training data for 64 neurons hidden layer size trained on a batch size of 1500*

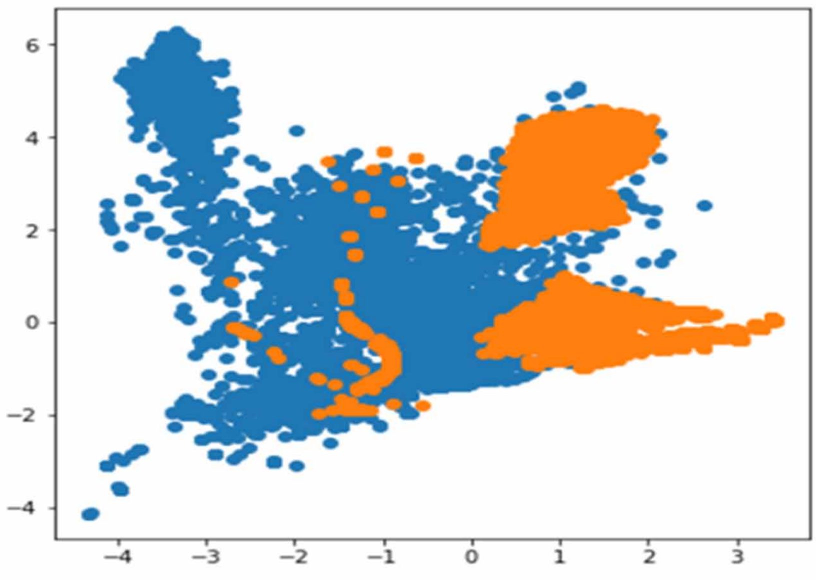


*Table 8. Confusion matrix results for second single 64-neuron hidden layer VAE-LDA*

| LDA Classification Confusion Matrix Results | |
|---|---|
| AUC(ROC) | 0.876576728986286 |
| Precision | 0.8761696784407927 |
| Recall | 0.9992155104785386 |
| F1 score | 0.933656016568543 |
| False positives | 11341 |
| True positives | 80244 |
| False negatives | 63 |
| True negatives | 34749 |

*Table 9. Confusion matrix results for second single 64-neuron hidden layer VAE-QDA*

| QDA Classification Confusion Matrix Results | |
|---|---|
| AUC(ROC) | 0.9166504976996729 |
| Precision | 0.9129945045567806 |
| Recall | 0.9992155104785386 |
| F1 score | 0.954161167195805 |
| False positives | 7647 |
| True positives | 80244 |
| False negatives | 63 |
| True negatives | 38443 |

The QDA produced 91.29% precision and 99.92% recall where LDA produced 87.62% precision and 99.92% recall.

A third model was trained with a batch size of 7500 and 64 neurons in the hidden layers of the encoder and decoder. The loss model is shown in Figure 20 and the visualisation is shown in Figure 21. The latent space visualization shows how DDoS flows occupy dense regions of the distribution. The precision of the QDA was

The overall performance of the VAE base QDA and LDA in terms of precision and recall was very high for the three models built. Given that no balancing of classes nor feature engineering was performed is good.

*Figure 20. Loss for a model with a single 64-neuron hidden layer VAE model (batch size = 7500)*

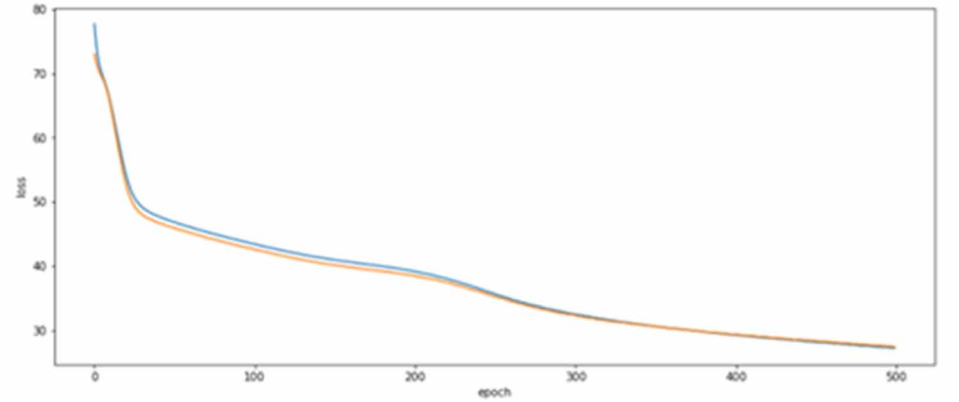


*Figure 21. Latent space visualization of training data for 64 neurons hidden layer size and batch size = 7500*

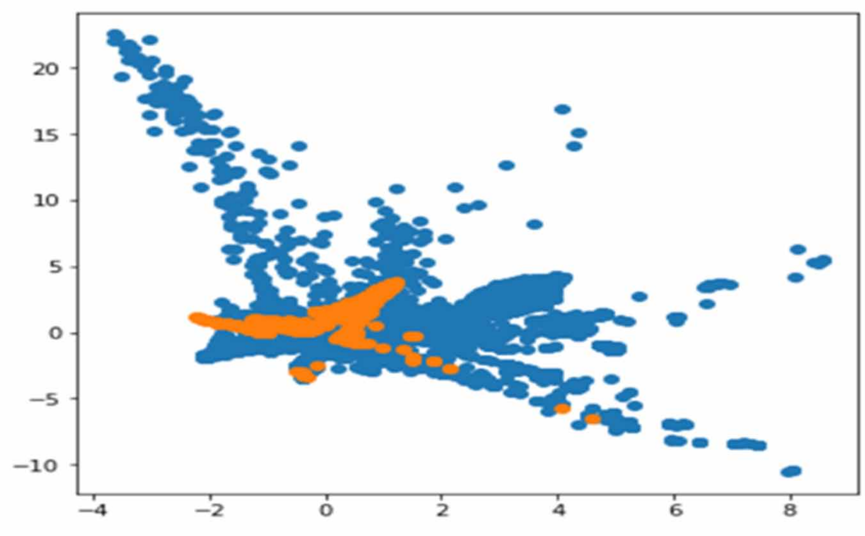


*Table 10. Confusion matrix results for third single 64-neuron hidden layer VAE-LDA, batch size = 7500*

| LDA Classification Confusion Matrix Results | |
|---|---|
| AUC(ROC) | 0.8794204086037665 |
| Precision | 0.8787055795871435 |
| Recall | 0.9991532494054067 |
| F1 score | 0.9350665998531656 |
| False positives | 11076 |
| True positives | 80239 |
| False negatives | 68 |
| True negatives | 35014 |

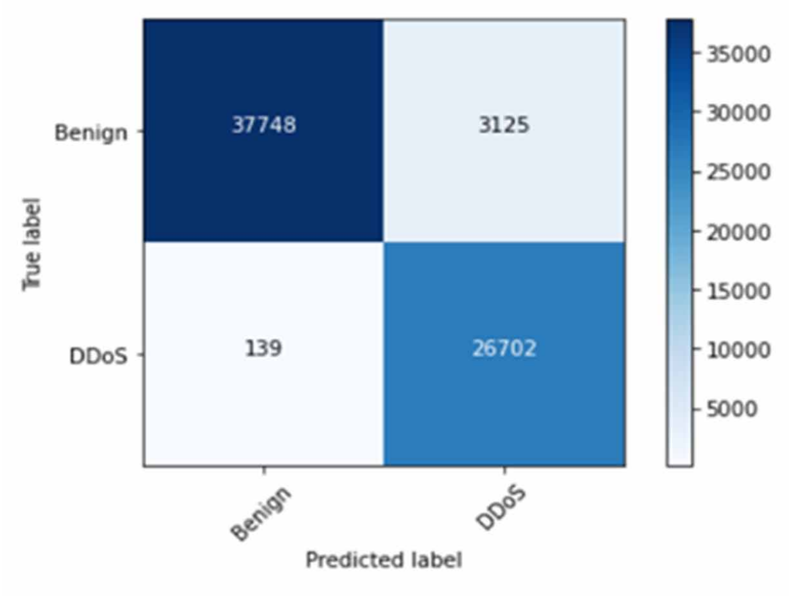
*Table 11. Confusion matrix results for third single 64-neuron hidden layer VAE-QDA, batch size = 7500*

| QDA Classification Confusion Matrix Results | |
|---|---|
| AUC(ROC) | 0.8658100310291411 |
| Precision | 0.8802057142857143 |
| Recall | 0.9590446660938647 |
| F1 score | 0.9179354854088327 |
| False positives | 10482 |
| True positives | 77018 |
| False negatives | 3289 |
| True negatives | 35608 |

### 4.2.2 Comparing Results of QDA Against Deep Neural Network

A Deep Neural Network was trained on the latent representation of the CICIDS 2017 Friday Afternoon data set network flows. The VAE model had a single hidden layer of 64 neurons and the coupled DDN had a single hidden layer of 512 neurons. The DNN did not train on the raw CICIDS 2017 dataset but got its input from the output of the VAE. The Assumption was the classes were already separated in the VAE latent space so it should be easier for downstream classifiers to separate the two classes. The Confusion matrix is shown in Table 12. The DDoS detector achieved a precision of 89.52% and a recall of 99.48%.

*Figure 22. Confusion matrix of VAE based DNN model, single 512-neuron hidden layer*



## Deep Neural Network 2 Hidden Layers

A second Deep Neural Network model was trained using two hidden layers of 32 and 16 neurons. The DNN was evaluated on the same test data used for evaluating the VAE-DA. The confusion matrix showing the performance of the DNN with 2 layers (16 and 32 neurons) is shown in Figure 23 and Table 12.

*Figure 23. Confusion matrix for DNN with 2 hidden layers of 16 and 32 neurons*

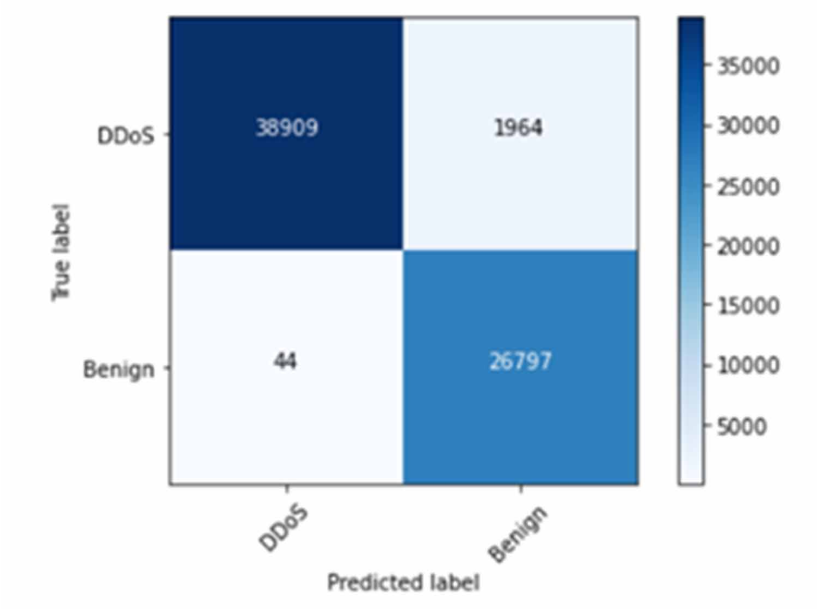


*Table 12. Confusion matrix results of DNN with 2 hidden layers of 16 and 32 neurons*

| DNN Classification Confusion Matrix Results | |
|---|---|
| Precision | 99.89% |
| Recall | 95.19% |
| False positives | 1964 |
| True positives | 77018 |
| False negatives | 44 |
| True negatives | 35608 |

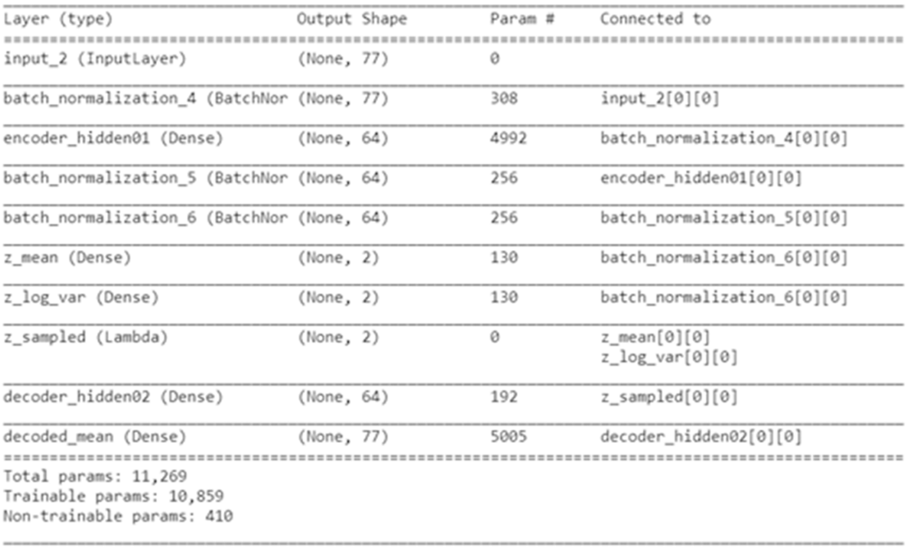
The DNN had 99.89% precision and a recall of 95.19%.

## VAE With a Single Hidden Layer Coupled to a DNN With Two Hidden Layers

A VAE was built with a single hidden layer and then coupled to a DNN with two hidden layers. The hidden layer of the encoder and decoder had 64 neurons each and the DNN had 32 neurons in each hidden layer. Both the VAE and DNN were trained with Keras Early Stopping configured to save the best model and its parameters during training (https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping, 2020).

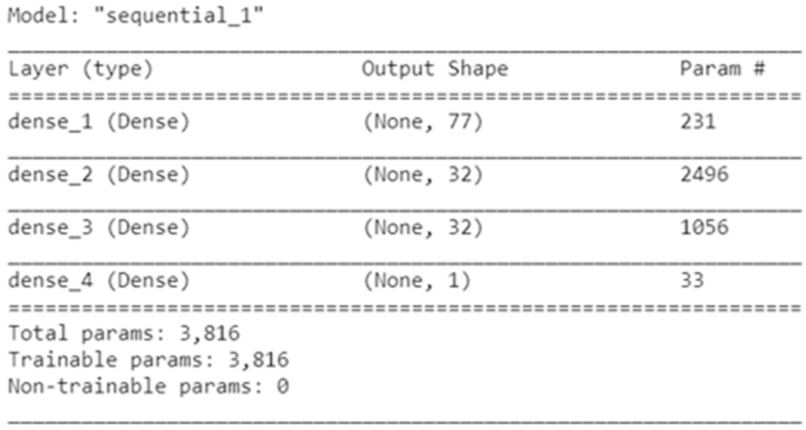
The Deep Neural Network model achieved 96.11% and 96.34% for precision and recall respectively.

*Figure 24. Summary of VAE model with a single 64-neuron hidden layer*



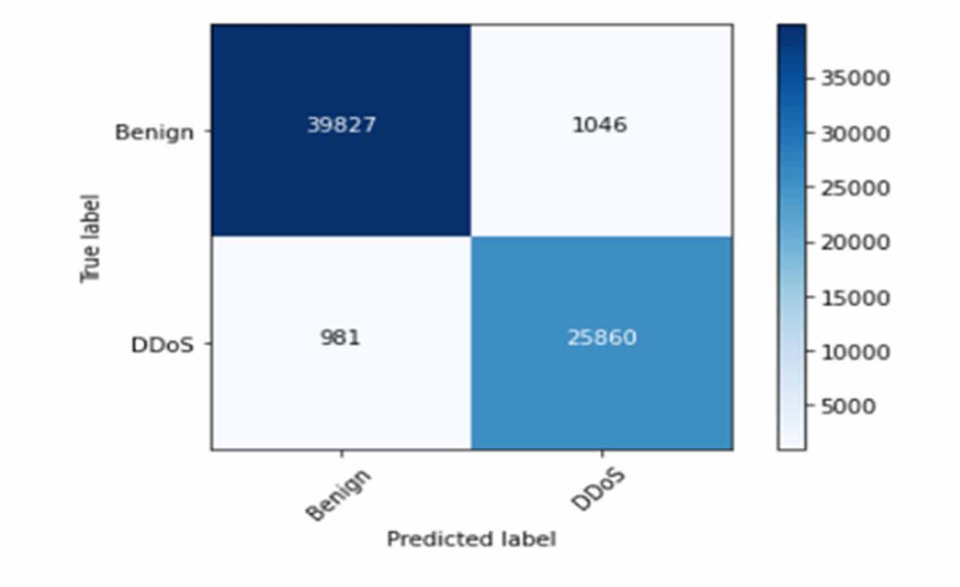
```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            (None, 77)           0
__________________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, 77)           308         input_2[0][0]
__________________________________________________________________________________________________
encoder_hidden01 (Dense)        (None, 64)           4992        batch_normalization_4[0][0]
__________________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, 64)           256         encoder_hidden01[0][0]
__________________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, 64)           256         batch_normalization_5[0][0]
__________________________________________________________________________________________________
z_mean (Dense)                  (None, 2)            130         batch_normalization_6[0][0]
__________________________________________________________________________________________________
z_log_var (Dense)               (None, 2)            130         batch_normalization_6[0][0]
__________________________________________________________________________________________________
z_sampled (Lambda)              (None, 2)            0           z_mean[0][0]
                                                                 z_log_var[0][0]
__________________________________________________________________________________________________
decoder_hidden02 (Dense)        (None, 64)           192         z_sampled[0][0]
__________________________________________________________________________________________________
decoded_mean (Dense)            (None, 77)           5005        decoder_hidden02[0][0]
==================================================================================================
Total params: 11,269
Trainable params: 10,859
Non-trainable params: 410
__________________________________________________________________________________________________
```

*Figure 25. Summary of DDN with two 32-neuron hidden layers*



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 77)                231
_________________________________________________________________
dense_2 (Dense)              (None, 32)                2496
_________________________________________________________________
dense_3 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 33
=================================================================
Total params: 3,816
Trainable params: 3,816
Non-trainable params: 0
_________________________________________________________________
```

*Figure 26. Confusion matrix for VAE-DNN with two hidden layers for DNN*



## 4.3 Analysis and Interpretation

Several models were built to come up with the best performing model of a VAE, QDA, LDA and DNN. A VAE model with a single 64-neuron hidden layer proved to be ideal for the complexity of the CICIDS 2017 Friday Afternoon data set. For the Deep Neural network two hidden layers of 32 neurons each produced the highest precision and recall ideal. The QDA proved to have higher precision and recall values than the LDA. Therefore subsequent comparisons with the Deep Neural Network involved the QDA only. The best VAE-QDA classifier showed higher precision and recall than traditional machine learning algorithm classifiers. The VAE-QDA performance was equally comparable to that of a Deep

Neural Network which had been trained on the same latent representation. The best VAE -QDA classifier achieved a precision and recall of 96.04% and 96.15% respectively. The best VAE-DNN model had 99.89% precision and a recall of 95.19%. A VAE-DNN with 96.11% and 96.34% for precision and recall respectively was not chosen because the cost of a False Positive is higher than that of a False Negative. The results supported the objective to build a DDoS classifier with high precision and recall values without feature engineering nor prior balancing of the classes within the data set.

For DDoS attack detection, both precision and recall needed to be very high, more so the precision. To avoid blocking or disconnecting benign traffic if the model were deployed in a real world environment precision needs to be maximized. A higher precision than recall may be preferable because allowing a few packets of DDoS packets usually does not harm servers whereas false positives usually negatively affect revenue. DDoS attack packets do not carry harmful payloads since the nature of the attack usually involves only overwhelming the victim server with normal packets to consume resources such as memory, CPU and network bandwidth of the Victim server. Since QDA assumed a quadratic decision boundary, it could accurately model a more complex range of classification problems than the LDA. The results prove that a DNN trained on the VAE latent representation of computer network flows performed better than QDA applied to the same VAE latent representation. The VAE-DNN had the additional advantage of scalability over QDA. Scalability is important for DDoS classifiers because the sizes of connected networks are ever expanding resulting in ever increasing amount of traffic. A classifier comprising VAE, QDA and DNN was finally built. The QDA and DNN served to endorse and confirm the results of tested flows.

### 4.4 Discussion

The VAE was able to reduce dimensionality from seventy seven to two. The DNN, QDA and LDA were able to learn from data with greatly reduced dimensions. The downstream classifiers scalability is improved as they can take in more data with reduced dimensionality. Because the QDA showed higher precision and recall values than the LDA the assumptions that go with QDA were proved true. It was therefore concluded that the benign and DDoS attack flows had different covariance matrix. The covariance matrix gave insight into the direction and scale of how the data is spread. QDA assumed that benign and DDoS attack samples had different covariance matrix. The visualisation of the latent space seemed to support the assumption that the benign and DDoS attack samples had different covariance matrix because the DDoS attack flows tended to be more clustered within the VAE probability distributions than benign traffic flows. The best DNN achieved precision and recall of 99, 89% and 95.19% respectively. The best QDA had a precision and recall of 96.04% and recall of 96.15% respectively. A VAE based DNN classifier was finally built because Deep Neural Networks have the advantage of scalability over Quadratic Discriminant Analysis classifiers.

## 5. CONCLUSION AND IMPLICATIONS

### 5.1 Conclusion

A way of training DDoS effective classifiers without the prior need to balance the data set nor feature engineering was demonstrated. The Variational Autoencoder (VAE), a generative model, proved that it can automatically learn the different classes' distribution and make classification easier for downstream classifiers. VAE based Deep Neural Network classifier is ideal to counter ever evolving nature of DDoS attacks. VAE based DNN is scalable to deal with an increasing number of connected devices and network traffic. The VAE was useful for greatly reducing the dimensionality of network flows from seventy seven to two. VAE based classifiers performed better than traditional machine learning classifiers. QDA classified network flows with higher precision and recall than LDA. Effective DDoS Classification models can be trained on unbalanced data without feature engineering by application of VAE based classification.

## 5.2 Implications

The application of QDA and LDA gave insight into how the network data met the assumptions that go with QDA and LDA. Because QDA classification results were consistently better than LDA it was concluded that the covariance matrix of DDoS attack samples differ from that of benign samples in the CICIDS 2017 data set. Detection of DDoS attacks and other intrusions can be made easier with the use of VAE-DNN. The VAE based DNN model scalability is ideal for modern day computer networks and the Internet of Things where huge amounts of network data are common. A VAE can be applied effectively where there is lack of domain knowledge since it learns features automatically. VAE can be used as a pre-processor of data to reduce dimensions for high dimensional data. With further research, unbalanced datasets may no longer compromise classifiers' precision and recall. By taking out the stages of data balancing and feature engineering in a data analytic pipeline model training times will be shortened. Because a trained VAE enhances the separation of classes, it makes it easier for downstream classifiers to achieve higher precision and recall values.

## REFERENCES

Alzahrani, A. H. (2018). Generation of DDoS Attack Dataset for Effective IDS Development and Evaluation. *Journal of Information Security, 9,* 225-241. doi:10.4236/jis.2018.94016

An, J., & Cho, S. (2015). *Variational Autoencoder based Anomaly Detection using Reconstruction Probability.* SNU Data Mining Center. Retrieved April 26, 2020, from http://dm.snu.ac.kr/static/docs/TR/SNUDM-TR-2015-03.pdf

Bellinger, C. D. (2017). Manifold-based synthetic oversampling with manifold conformance estimation. *Machine Learning*.

Bonguet, A., & Bellaiche, M. (2017). A Survey of Denial-of-Service and Distributed Denial of Service Attacks and Defenses in Cloud Computing. *Future Internet*, *9*(3), 43. Advance online publication. doi:10.3390/fi9030043

Carl, D. (2016). Tutorial on Variational AutoencodersCarnegie Mellon UC Berkeley.

Darko Galinec, D. M. (2017). Cybersecurity and cyber defence: National level strategic approach. *Journal for Control, Measurement, Electronics, Computing and Communications*.

Dietrich, D., Heller, B., & Yang, B. (2015). *Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. John Wiley & Sons, Inc.

Doersch, C. (2016, August 13). Tutorial on Variational Autoencoders. https://www.researchgate.net/publication/304163568

Elmi, A. H., Sallehuddin, R., Ibrahim, S., & Zain, A. M. (2014). Classification of SIM Box Fraud Detection Using Support Vector Machine and Artificial Neural Network. *International Journal of Innovative Computing*, 19–27.

Fajardo, A. V. (2018). VOS: A Method for Variational Oversampling of Imbalanced Data. *International Journal of Advanced Computer Science and Applications*, 8.

Ghojogh, B., & Crowley, M. (2019). *Linear and Quadratic Discriminant Analysis: Tutorial*. Machine Learning Laboratory, University of Waterloo.

Gupta, A. (2019). *Generative Image Translation for Data Augmentation of Bone Lesion Pathology. Proceedings of Machine Learning Research, 102, 225–235* .

Ikeda, Y. (2018, December 21). Estimation of Dimensions Contributing to Detected Anomalies with Variational Autoencoders. arXiv:1811.04576v2 [stat.ML].

Jiadong, R. J., & Wang, Q. (2019, June 16). Building an Effective Intrusion Detection System by Using Hybrid Data Optimization Based on Machine Learning Algorithms. *Hindawi Security and Communication Networks*, 11.

Jiayu, S. X. (2018). Learning Sparse Representation With Variational Auto-Encoder for Anomaly Detection. *IEEE Open Access Journal*.

Jorge, J. V. (2018). *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018)* - Volume 5 (pp. 96-104). Science and Technology Publications.

Khan, R. S. (2019). *An Adaptive Multi-Layer Botnet Detection Technique Using Machine Learning Classifiers*. doi:10.3390/app9112375

Kim, J. (2020, April 1). Botnet Detection Using Recurrent Variational Autoencoder. doi:10.1109/GLOBECOM42002.2020.9348169

Kingma, D. W. (2019). An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning, 12*(4), 307-392. doi:10.1561/2200000056

Kurien, L. C. (2019, December). An Ameliorated method for Fraud Detection using Complex Generative Model: Variational Autoencoder. *International Journal of Innovative Technology and Exploring Engineering*, *9*(2S).

Lopez-Martin, M. E. (2018, December). Variational data generative model for intrusion detection. *Knowledge and Information Systems*. Advance online publication. doi:10.100710115-018-1306-7

Macfarlane, R. B. (2015). TFTP DDoS amplification attack. *Computers & Security*. Advance online publication. doi:10.1016/j.cose.2015.09.006

Mayuranathan, M. (2019). Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment. *Journal of Ambient Intelligence and Humanized Computing*. Advance online publication. doi:10.100712652-019-01611-9

Mitra, A. (2017, March 2). *What is smurf attack*. Retrieved May 08, 2020, from https://www.thesecuritybuddy.com/dos-ddos-prevention/what-is-smurf-attack/

Noble, G., & Sujitha, M. (2015). Flow Based Solutions for DoS and DDoS Attack Detection. *International Journal of Advanced Research in Computer Engineering and Technology*.

Reaves, B., Shernan, E., Bates, A., & Carter, H. (2015). Boxed Out: Blocking Cellular Interconnect Bypass Fraud at the Network Edge. *Proceedings of the 24th USENIX Security Symposium*.

Rezvy, S. P. (2019). Intrusion Detection and Classification with Autoencoded Deep Neural Network. *11th International Conference, SecITC 2018.* 10.1007/978-3-030-12942-2_12

Riaz, U. K., Xiaosong, Z., Rajesh, K., Abubakar, S., A, N. G., & Mamoun, A. (2019). An Adaptive Multi-Layer Botnet Detection Technique Using Machine Learning Classifiers. *Applied Sciences*.

Sallehuddin, R., Ibrahim, S., Mohd Zain, A., & Hussein Elmi, A. (2015, November 26). Detecting SIM Box Fraud by Using Support Vector Machine and Artificial. *Jurnal Teknologi*, *74*(1), 3. doi:10.11113/jt.v74.2649

Sharafaldin, I. L. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *ICISSP 2018 - 4th International Conference on Information Systems Security and Privacy.* SCITEPRESS – Science and Technology Publications, Lda. 10.5220/0006639801080116

Singh, K. S. (2015). A systematic review of IP traceback schemes for denial of service attacks. Punjab: Department of Computer Science and Engineering, S B S State Technical Campus, Punjab 152004, India. doi:10.1016/j.cose.2015.06.007

Tharwat, A. G. (2017). *Linear Discriminant Analysis: A Detailed Tutorial.* dx.doi.org/10.3233/AIC-170729

Vipin, A. (2018). Analysis and detection of SIM box. *International Journal of Advance Research, Ideas and Innovations in Technology, 4*(3).

Wang, W. W. (2018). A HMM-R Approach to Detect L-DDoS Attack Adaptively on SDN Controller. *Future Internet MDPI, 10*(83). doi:10.3390/fi10090083

Wanga, X., Dua, Y., Linb, S., Cuia, P., Shenc, Y., & Yanga, Y. (2019, November 17). adVAE: a Self-adversarial Variational Autoencoder with Gaussian Anomaly Prior Knowledge for Anomaly Detection. arXiv:1903.00904v3.

Wangy, Z. (2017). DDoS Event Forecasting using Twitter Data. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17).*

Yang, Y. (2019). Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational Auto Encoder and Deep Neural Network. *Sensors*, 2.

Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2019, June 2). Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network. *Sensors (Basel)*, *19*(11), 2528. doi:10.339019112528 PMID:31159512

Yang, Z., Zheng, K., Wu, C., & Yang, Y. (2019). Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network. *Sensors (Basel)*, *19*(11), 2528. Advance online publication. doi:10.339019112528 PMID:31159512

Yilmaz, I. M. (2019). Expansion of Cyber Attack Data From Unbalanced Datasets Using Generative Techniques. arXiv:1912.04549v1 [cs.LG].

Yulianto, A. S. (2018). Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset. In *The 2nd International Conference on Data and Information Science*. IOP Publishing. 10.1088/1742-6596/1192/1/012018

Zhang, C. S. (2016). An Imbalanced Data Classification Algorithm of De-noising Auto-Encoder Neural Network Based on SMOTE. *MATEC Web of Conferences, 56*. 10.1051/matecconf/20165601014