Web Application Attack Detection Using Deep Learning

Tikam Alma
Dhirubhai Ambani Institute Of Information and Communication Technology
Gandhinagar, India
201601030@daiict.ac.in

Manik Lal Das
Dhirubhai Ambani Institute Of Information and Communication Technology
Gandhinagar, India
maniklal_das@daiict.ac.in

***Abstract*—Hypertext transfer protocol (HTTP) is the core communication protocol used to access the internet i.e www and is used by all of today's web application.All HTTP messages (Requests and Responses) consist of one or more headers and these are the fields where most of the exploits and payloads were injected to do the web attacks.According to the OWASP-Top 10 the 80 percent of the web attacks were done through HTTP/HTTPS requests queries that include methods like GET, POST and PUT.HTTP request parameters are intercepted or captured by attackers and these parameters are altered to inject attack vectors and payloads to exploit the endpoint of the systems.We propose a deep learning based web attack detection engine to detect benign and anomaly queries and also to classify which type of attack it is by classification engine.Web application attack detection engine which is trained on 40,000 anomaly and benign web queries to achieve the accuracy of receiver operating characteristic curve of 1,the ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.**

***Index Terms*—Information Security, Deep-Learning, Web Application Security**

I. Introduction

Web attacks are the most common and most dangerous attacks, that exclusively use the HTTP/HTTPS protocols. Symantec Internet security Threat Report (ISTR-24)[8] reveals statistics of 1 in 10 URLs analyzed were identified as being malicious, in 2018, overall web attacks on endpoint systems increased by 56 percent in 2018.

Among web attacks, formjacking was one of the biggest cyber security trend of the year,with an average of 4800 websites every month in 2018.According to Imperva Web Application Vulnerability Report 2019,[10] high severity attacks are injection attacks which are exploited through injecting payloads in HTTP/HTTPS web queries by using GET, POST and PUT request and responses methods of the HTTP protocol, namely, CSRF (Cross Site Request Forgery) attack, SQL Injection attack, XSS (Cross Site Scripts) attacks, and widely used Vulnerable JS libraries , which account for 51 percent, 27 percent, 33 percent and 36 percent respectively.

This paper focuses on the most frequent types of web based injection attacks which includes SQL Injection, XSS(Cross Site Script), RFI (Remote File Inclusion), XXE (XML External Entity), CSRF (Cross Site Request Forgery), SSRF (Server Side Request Forgery) of the the entire web attacks.

Intrusion detection system IDS are system of frameworks that monitor the network traffic in web applications.Web IDS acts as intermediate between web application and users,it analyzes web traffics to detect any anomaly or malicious activity.[2]

Currently there are two types of detection approaches are used anomaly based detection and signature based detection.Signature based IDS are based on looking for "known patterns" of detrimental activities.Similar to antivirus, after a new attack is recorded the data files need to be updated before the network is secure.Signature based IDS has low alarm rates, all it has to do is look up the list of known signatures of attacks and if it finds a match report it.The negative point of signature based attack is, if someone develops a new attack,there will be no protection.

Anomaly based detection are based on tracking unknown unique behaviour patterns of detrimental activities.It has a disadvantage,but in this proposed paper we come with a solution, for that problem that is, False positive, catches too much because behavioural based IDS monitor a system based on their behavior pattern.

Anomaly based detection approaches[2] usually rely on a detection model to identify anomalous web requests.In, contrast to signature based approaches,anomaly based approaches can detect unknown attack,but with a high False positive rate,in this paper we solve the problem of False positive and Negative Rates by using Machine Learning algorithms to built a efficient detection model.

After an anomaly has been detected, it may become a signature.There are Adaptive and Constant Detection further which comes under Anomaly Detection[4].Adaptive Detection Method always incorporate new traffic data to induce an up-to-date detection model.While Constant detection method use a period of traffic data to build a detection model once and use it for all later detection

Adaptive Detection : These detection method take network traffic of port 80, that is HTTP traffic as data source and analyze HTTP packet payloads. Static based - a method which

uses static based methods to develop an adaptive blacklist based packet filter,which can help filter out network based on IP confidence.Dynamic Clustering - Dynamic Clustering to detect HTTP attacks adaptive.Constant Detection : constant detection method are widely applied to intrusion detection,but they suffer from high False Positive Rate.[2]
The Traditional patching approach to mitigating most network layer vulnerabilities just doesn't work the same with web application vulnerabilities, such as SQLi, XSS or Remote Code Execution.This is due to the fact that web application are poorly designed and insecure infrastructure coding, during the development and deployment phase,must follow owasp secure coding guidelines to prevent most of the attacks, all we can do it to continuously detect the attack against the web application to prevent and secure it as early as possible.
We need an adaptive detection model, which can once detect anomalies and classify them which type of attack it is so that the developer at back-end can specifically fix that patch or prevent it bu secure coding methodologies, for example Django uses CSRF Tokens in the framework ti prevent CSRF web attacks which account for 51 percent of the web attack according ti the Symantec ISTR-24, 2019 Report.
Also, we need adaptive detection model, which can learn the patterns overtime to detect unknown web attacks and identify which type of attack vectors are being exploited.Because the on the other side of web, that is hackers are actively scanning and doing reconnaissance for vulnerabilities and flaws, updating attacks vectors with different encoding techniques, for example latest injection attack technique is polyglots, polyglots are encoded attacks vectors on different algorithms, it will execute on JavaScript and HTML but the IDS and polyglots will never detect it, because it has never seen it that type of patterns.Using pattern recognition of these string and character based patterns, we can detect more accurately anomalies before it's requests going to the server.
There are several reasons why conventional Intrusion Detection System or WAFs do not work:

- Limited Dataset - To collect and capture large amount of anomalous data, we need to set-up a system or an automated system that captures the attack requests and classify them whether they are anomalous or normal requests, more like a attack-defense simulation system,but smart enough to classify, that does not need to be labeled manually and it will save lots of time.Because there is no such system or software we trained our detection model on existing labeled data.
- High False Positive - Although prior work has applied unsupervised learning algorithms such as PCA[5] and SVM[12] to detect web attacks, these approaches require manual selection of attack specific features.[6]These conventional method may achieve acceptable performance but, they have high false positive rates.A 1 percent in false positive may cause intrusion detection system to incorrectly flag 1000s of legitimate users.
- Labeled Data - Many Intrusion detection system rely on rule-based strategies or supervised machine learning algorithms to differentiate normal requests from attack requests,which require large amounts of labeled training data to train the learning algorithms[6].We trained our classification model on 7 different types of attacks with minimum amounts of labeled data-set with 60 percent accuracy.

We approach all these problem with applying deep learning algorithms to detect web application attacks autonomously in real-time with classification model, that is efficient, secure and very fast.Our approach applies deep learning to to implement detection model and classification model,from feature engineering to prediction.
Our proposed architecture first uses 40,000 web request which were anomaly and benign for training and then 20,000 anomaly web requests and responses also for training and testing.Then a classification engine which is trained on ECML-KDD dataset for classification of anomaly queries, which type of attack it is.
We implemented the detection model on sequence to sequence model,which consist of encoder and decoder, which sets its target values equal to its input values. Auto-encoder is sequence to Sequence model, which is an architecture of Recurrent Neural Network, which doesn't need labeled data, it can learn from the sequences of word and weight each word or character according to them,and auto-encoder re-creates things or sequences that it has seen,in other words, approximate the identity function.
In detection model if we give input an anomalous sample requests, it is likely to re-create it with a high-degree of errors, because of never having seen such a sample previously.If the requests sample is normal the Loss Value is Low as possible,Lower Than 1.If the requests sample were anomalous the loss value will be greater than 1.

The paper is organized as follows,Section 2 summarizes,the background and related works,existing works and bookmarks.Section 3 describes the system design and implementation.Section 4 empirically evaluates the performance of our detection and classification model.Section 5, presents concluding remarks.

II. Background

A. Deep Learning Approach to Web Attack Detection

Machine Learning approaches for detecting web attacks can be categorized into two types: supervised and unsupervised. Supervised Learning : Approaches such as Naive bayes[7] and SVM[1],supervised learning is typically done in the context of classification, when we want to map input to output labels, or regression, when we want to map input to a continuous output. Supervised learning is where input variables (X) and an output variable (Y) is used in an algorithm to learn the mapping function from the input to the output

$$Y = f(X)$$

If web attacks labeled dataset is trained using supervised algorithms such as SVM and Naive Bayes, then it will just classify anomalous to Normal Web Attack requests, but the problem model will get is, it cannot handle new types of attack requests and secondly the model will be limited to a certain amounts, because it will need a large amount of labeled dataset.

Unsupervised Learning: Approaches to machine learning web attack detection such as Principal Component Analysis (PCA)[11] and Auto-encoders[9] do not require labeled training dataset. Unsupervised Learning is very useful in exploratory analysis because it can automatically identify patterns and structure's in data. Dimensional reduction, which refers to the methods used to represent data using less columns or features, can be accomplished through unsupervised method. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

PCA computes eigenvectors of the co-variance matrix ("principal axes") and sorts them by their eigenvectors (amount of explained eigenvectors).The centered data can then be projected onto these principal axes to yield principal component("scores"), for the purpose of dimentionality reduction. Let X_{raw} be the n x p data matrix with n rows (data points) and p columns (variables or features).After subtracting the mean U from each row, we get the centred data matrix X. Let V be the p x k matrix of some K eigenvectors that we want to use; these would most often be two k eigenvectors with the largest eigenvectors.Then the n x k matrix of PCA projects ("Scores") will be simply given by

$$Z = XV$$

.

In order to be able to reconstruct the original two variables from this one principal component, we can map it back to p dimensions with V^t.The values of each PC should be placed on the same vectors as was used for projection; the result is then given by:

$$X^{`} = ZV^t = XVV^t$$

PCA Reconstruction = PC Score * EigenVectors(t)+Mean

If all p eigenvectors are used then VVt is the identity matrix, no dimensionalty reduction is performed,hence "reconstruction" is perfect.

PCA is restricted to a linear map,while autoencoders can have non linear encoder/decoders.A single layer autoencoder with linear function is nearly equivalent to PCA, where nearly means that the W found by Autoencoder and PCA won't be the same , but the subspace spanned by the respective W's will.

For a dataset involving a huge number of features,be it image,text or video data,we can't apply regular machine learning algorithms models directly onto it,we need some prepossessing step which will reduce the training time but still not reduce representation power of data that is accuracy /errors trade-off must be less.

For the detection model we have used Sequence-to-Sequence autoencoder because,an autoencoder is special type of feed forward neural network which encodes input X into hidden layer h and then decodes it back from its hidden representation.The model is trained to minimize the loss between the input and the output layer, the hidden layer will able to reconstruct Xcap perfectly, h is lossless encoding xi.It captures all important characteristics of xi.Here, h behaves like PCA's reduced dimension matrix from which the output is reconstructed with some loss in value.Hence the encoder part shows resemblance to PCA.

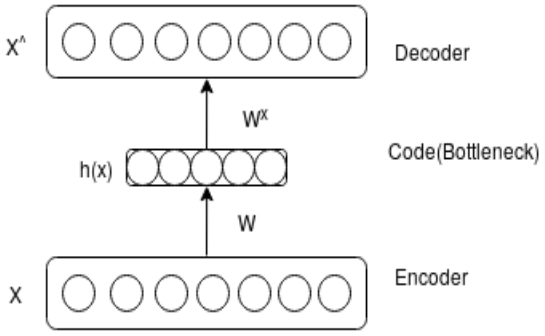

Fig. 1. Autoencoder basic architecture

Here are the condition that makes autoencoder an PCA,encoder part will be equivalent to PCA if linear encoder, linear decoder, square error loss function with normalized inputs are used. Which means PCA is restricted to linear maps only whereas autoencoders are not. Due to these linearity constraints, we moved to encoders with sigmoid-like non-linear functions which give more accuracy in the reconstruction of data.

Due to these linearity constraints,we moved to encoders with Sigmoid-like non-linear functions which give more accuracy in the reconstructions of the data.

Autoencoders are closely related to principal component analysis (PCA). In fact, if the activation function used within the autoencoder is linear within each layer, the latent variables present at the bottleneck (the smallest layer in the network, aka. code) directly correspond to the principal components from PCA. Generally, the activation function used in autoencoders is non-linear, typical activation functions are ReLU (Rectified Linear Unit) and sigmoid.

The math behind the networks is fairly easy to understand, so I will go through it briefly. Essentially, we split the network into two segments, the encoder, and the decoder.

$$\Phi : \chi \to F \tag{1}$$

$$\Psi : F \to \chi \tag{2}$$

$$\Phi, \Psi = argmin_{\Phi,\Psi} ||X = (\Phi * \Psi)X||^2 \quad (3)$$

The encoder function, denoted by Φ, maps the original data X, to a latent space F, which is present at the bottleneck. The decoder function, denoted by Ψ, maps the latent space F at the bottleneck to the output. The output, in this case, is the same as the input function. Thus, we are basically trying to recreate the original image after some generalized non-linear compression.

The encoding network can be represented by the standard neural network function passed through an activation function, where z is the latent dimension.

Number of nodes in any hidden layer,Number of hidden layers applicable for deep autoencoders,Activation unit such as sigmoid, tanh, softmax, and ReLU activation functions,the rectified linear activation function is a piecewise linear function that will output the input directly if is positive, otherwise, it will output zero,Regularization parameters or weight decay terms on hidden unit weights

$$Z = \sigma(W_x + b) \quad (4)$$

Similarly, the decoding network can be represented in the same fashion, but with different weight, bias, and potentially activation functions being used.

$$X^{'} = \sigma^{'}(W^{'} z + b^{'}) \quad (5)$$

The loss function can then be written in terms of these network functions, and it is this loss function that we will use to train the neural network through the standard backpropagation procedure.

$$L(x, x^{'}) = ||x - x^{'}|| = ||x - \sigma^{'}(W^{'}(\sigma(Wx + b)) + b^{'})||^2 \quad (6)$$

Since the input and output are the same images, this is not really supervised or unsupervised learning, so we typically call this self-supervised learning. The aim of the autoencoder is to select our encoder and decoder functions in such a way that we require the minimal information to encode the image such that it be can regenerated on the other side.

III. System Architecture and Implementation

Detection Engine:

For the implementation of detection engine we have used Autoencoder model which Uses Sequence to Sequence architecture which is completely made up of LSTM(Long Short Term Memory)[3] cells.

An LSTM Autoencoder is an implementation of autoencoder for sequence data using an Encoder-Decoder LSTM architecture.Autoencoder as previously explained are type of Self-supervised learning model that can learn a compressed representation of input data.Sequence prediction problems are challenging, not least because the length of the input sequence can vary.

Recurrent neural networks,such as the Long Short Term Memory or LSTM, networks are specifically designed to support of input data.They are capable of learning the complex dynamics within the temporal ordering of input sequences as well as internal memory to remember or use information across long input sequences.The LSTM network can be organized into an architecture called the Encoder-Decoder LSTM that allows the model to be used to both support variable length input sequences and to predict or output variable length output sequences.

In this architecture, an encoder LSTM model read the input sequences step-by-step.After reading in the entire sequences,the hidden state or output of this model represents an internal learned representation of the entire input sequences as a fixed-length vector.This vector is then provided as an input to the decoder model that interprets it as each step in the output sequences is generated.

For a given dataset of sequences, an encoder-decoder LSTM is configured to read the input Sequence, encode it, decode it, and recreate it.The performance of the model is evaluated based on the model's ability to recreate the input sequences.

For Detection model we are giving Raw HTTP requests,which has lots of characters numbers and strings in Sequence form, as input for training these inputs are processed in sequence to sequence LSTM cells.

The Detection model and Classification model will work in synchronization as follows:

1) For the training purpose,large amounts of Unlabeled normal HTTP requests were collected from open-source Vulnbank organization,which contains 40k normal HTTP [GET,POST and PUT] methods requests.
2) For the autoencoder's [Encoder-Decoder] architecture the Hyperparameters were set.These hyperparameters can be trained by setting the problem as a grid search problem. However, each hyperparameter combination requires training the neuron weights for the hidden layer(s), which results in increasing computational complexity with an increase in the number of layers and number of nodes within each layer. To deal with these critical parameters and training issues, stacked autoencoder concepts have been proposed that train each layer separately to get pretrained weights, and then the model is fine-tuned using the obtained weights. This approach tremendously improves the training performance over the conventional mode of training. Number of nodes in any hidden layer,Number of hidden layers applicable for deep autoencoders,Activation unit such as sigmoid, tanh, softmax, and ReLU activation functions,the rectified linear activation function is a piecewise linear function that will output the input directly if is positive, otherwise, it will output zero,Regularization parameters or weight decay terms on hidden unit weights.For our architecture we implemented with:

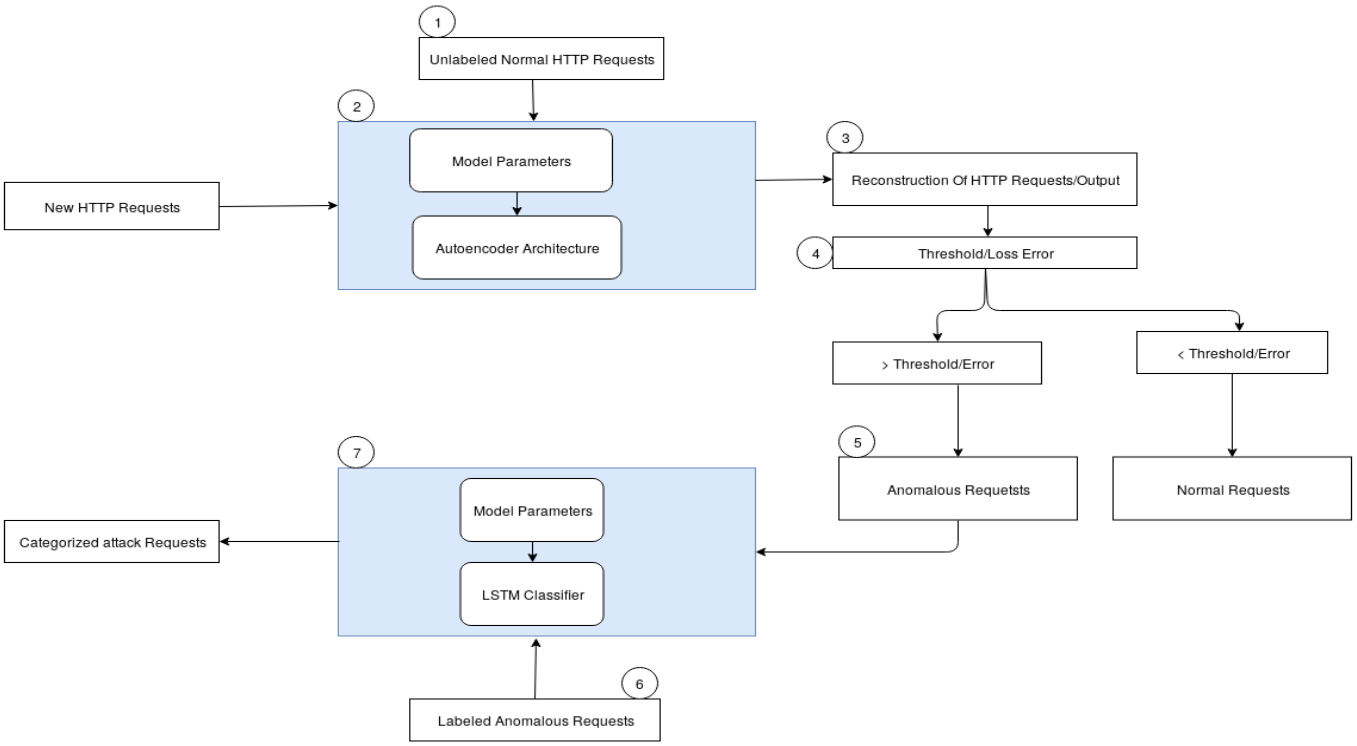

Fig. 2. System Architecture

- Batch Size = 128
- Embed Size = 64
- Hidden Size = 64
- Number of Layers = 2
- Dropout Rate = 0.7

3) Reconstruction of requests were done by the decoder, that is $X^{'} = \sigma^{'}(W^{'} + b^{'})$.Which is perfectly reconstruction of the given input and is evaluated Loss Function and accuracy.
4) When a new requests is given input to the trained autoencoder,it will decode and encode,the requests vector and calculate the reconstruction or Loss Error.If Loss error is larger than the learned threshold θ,it will categorized as anomalous requests.If loss error is smaller then θ,it will be categorized as normal requests.
5) After categorizing requests into Normal and Anomalous requests,Normal requests were sent to the database for retraining or re-learning so that over time the detection model will learn new type of requests patterns.Anomalous requests were to the classification model that will further categorize the anomalous requests into which type of attack it was exploited through requests like SQLi, XSS or CSRF etc.
6) The classification model is trained on larger number of labeled attack vectors HTTP requests.It contains 7 class of attacks which are Os-Commanding, PathTraversal, SQL-injection, X-PathInjection, LDAPInjection, SSI, and XSS.
7) We used LSTM layers to train the classification model,and fine tuned the model with hyperparametrs.Every LSTM layer should be accompanied by a Dropout layer. This layer will help to prevent overfitting by ignoring randomly selected neurons during training, and hence reduces the sensitivity to the specific weights of individual neurons. 20 percent is often used as a good compromise between retaining model accuracy and preventing over-fitting. Hyperparameters were set as follows:

```
GET /"><img src="javascript:alert(document.domain)">? HTTP/1.0
Host: 244.33.234.83
Connection: keep-alive
Accept: application/*;q=0.0, text/xml
Accept-Charset: *;q=0.1
Accept-Encoding:
Accept-Language: eb-eiprhd;q=0.4, 3Keietoa-hee;q=0.6, eyb0alal-t
Cache-Control: min-fresh=967
```

Fig. 3. Raw HTTP Requests with XSS attack Vector

The image in figure-3 is the RAW anomaly http requests with XSS attack vector, in data pre-processing step,the Raw HTTP data is converted to a single string and parsed as

input to the LSTM cell,which is the passed to the training algorithm to train the model.

IV. Experimental Results and Evaluation

We Evaluated our Detection model on ROC curve.An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters 1) True Positive Rate and 2) False Positive Rate. The term anomaly suggests an event that stands out from the rest.Because of the difficulty of reliably defining normality with a descriptive feature set, anomalies raised by systems can sometimes be fraught with false alarms(False Positives) or missed alerts(False negatives). False negatives occurs when the system does not find something that the users intend it to find it.False positives occur when the system erroneously recognizes normal events as anomalous ones.

Here we used ROC curve to evaluate the architecture's results,An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds.Test accuracy; the closer the graph is to the top and left-hand borders, the more accurate the test. Likewise, the closer the graph to the diagonal, the less accurate the test. A perfect test would go straight from zero up the the top-left corner and then straight across the horizontal.
1)A false positive (FP)or false alarm,which refers to the detection of benign traffic as an attack,and (2)A false negative (FN), which refers to detecting-attack traffic as benign traffic. A key goal of an intrusion detection system should be to minimize both the FP rate and FN rate. A trade-off exists, however, since a more strict algorithm will tend to reduce the FN rate, but at the cost of detecting benign traffic as attack traffic. **TP** True Positive the number of observations correctly assigned to the positive class
FP False Positive the number of observations assigned by the model to the positive class.

TPR (True Positive Rate) reflects the classifiers ability to detect members of the positive class (pathological state)

$$TPR = \frac{TP}{(TP + FN)}$$

FPR (False Positive Rate) reflects the frequency with which the classifier makes a mistake by classifying normal state as pathological.

$$FPR = \frac{FP}{(FP + TN)}$$

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.

Results were as follow:

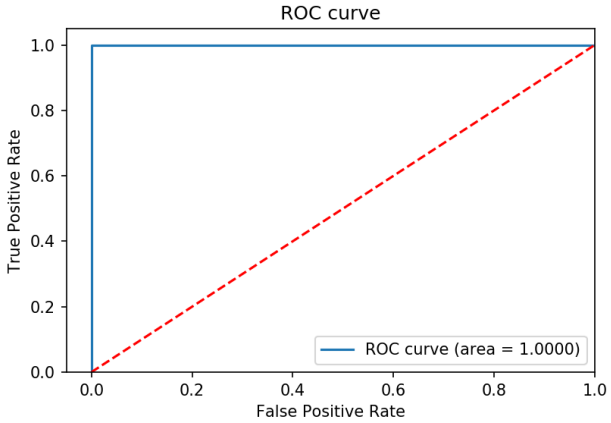

Fig. 4. ROC Curve

- Precision : 0.9979
- Recall : 1.00

True Postive and False Postive:

- Number of True Positive : 1097
- Number of Samples : 1097
- True Positive Rate : 1.00
- Number of False Positive : 7
- Number of samples : 2200
- False Positive Rate : 0.0032

V. Conclusion

This paper describes the architecture and the results of applying deep-learning models to the Information Security domain for anomaly detection and classification.We implemented a detection engine and classification model which will work simultaneously to detect and categorize the type of anomalous attack vector requests.The core concept and mathematical part of the autoencoder is simple Principal Component Analysis(PCA),which was implemented on neural network to do complex analysis and reconstruction of the input data.Because of less amount of labeled categorized anomalous dataset classification engine is not 100 percent accurate it can be improved with optimized training with good amounts of dataset in future work.

References

[1] C. Cortes and V. Vapnik, Support vector machine,Machinelearning, vol. 20, no. 3, pp. 273297, 1995.
[2] Ying Donga and Yuqing Zhanga, Adaptively Detecting Malicious Queries in Web Attacks, https://arxiv.org/pdf/1701.07774.pdf
[3] Sepp Hochrieter and Jurgen Schmidhuber, Neural Compuatation,1997.
[4] Kenneth L. Ingham Hajime Inoue,Comparing Anomaly Detection Techniques for HTTP
[5] G. Liu and Z. Yi and S. Yang A hierarchical intrusion detectionmodel based on the pca neural networks,Neurocomputing, vol. 70,no. 7, pp. 15611568, 2007
[6] Pan, Y., Sun, F., Teng, Z. et al. Detecting web attacks with end-to-end deep learning. J Internet Serv Appl 10, 16 (2019) doi:10.1186/s13174-019-0115-x.
[7] X. Xu and X. Wang,An adaptive network intrusion detectionmethod based on pca and support vector machines,AdvancedData Mining and Applications, pp. 731731, 2005

[8] S. Russell and P. Norvig and A., Intelligence, A modern approach,Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs, vol. 25, p. 27,1995.
[9] Symantec Internet Security Threat url=https://cdn2.hubspot.net/hubfs/4595665/Acunetix_web_application_vulnerability_report_2019.pdf
[10] Pascal Vincent and Hugo Larochelle,Stacked Denoising Autoencoders: Learning Useful Representationsina Deep Network with a Local Denoising Criterion,Journal of Machine Learning Research 11 (2010).
[11] Web Application Vulnerability Report2019 url=https://cdn2.hubspot.net/hubfs/4595665/Acunetix_web_application_vulnerability_report_2019.pdf
[12] S. Wold and K. Esbensen and P. Geladi, Principal component analysis,Chemometrics and intelligent laboratory systems, vol. 2, no. 1-3,pp. 3752, 1987
[13] Classification of HTTP Attacks: A Study on theECML/PKDD 2007 Discovery ChallengeBrian Gallagher and Tina Eliassi-RadLawrence Livermore National Laboratory.
[14] Towards Two-Dimensional Sequence to Sequence Model in NeuralMachine TranslationParnia Bahar, Christopher Brix and Hermann NeyHuman Language Technology and Pattern Recognition GroupComputer Science Department