TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO: 073/MSCS/653

User Behavior Modeling and Anomaly Detection in Cybersecurity Data Using Deep Learning

By

Balaram Sharma

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SYSTEM AND KNOWLEDGE ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

NOVEMBER, 2019

User Behavior Modeling and Anomaly Detection in Cybersecurity Data Using Deep Learning

by

Balaram Sharma
073/MSCS/653

Thesis Supervisor
Dr. Basanta Joshi

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer System and Knowledge Engineering

Department of Electronics and Computer Engineering Institute of Engineering,
Pulchowk Campus
Tribhuvan University
Lalitpur, Nepal

November, 2019

COPYRIGHT

DECLARATION

I declare that the work hereby submitted for Master of Science in Computer System and Knowledge Engineering (MSCSKE) at IOE, Pulchowk Campus entitled "User Behavior Modeling and Anomaly Detection in Cybersecurity Data Using Deep Learning" is my own work and has not been previously submitted by me at any university for any academic award. I authorize IOE, Pulchowk Campus to lend this thesis to other institution or individuals for the purpose of scholarly research.

Balaram Sharma
073/MSCS/653

Date: November, 2019

RECOMMENDATION

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering for acceptance, a thesis entitled **"User Behavior Modeling and Anomaly Detection in Cybersecurity Data Using Deep Learning"**, submitted by Balaram Sharma in partial fulfillment of the requirement for the award of the degree of **"Master of Science in Computer System and Knowledge Engineering"**.

.....

Supervisor: Dr. Basanta Joshi,
Assistant Professor,
Department of Electronics and Computer Engineering,
Institute of Engineering, Tribhuvan University.

.....

External Examiner: Mr. Om Bikram Thapa
Chief Technology Officer,
Vianet Communications.

.....

Committee Chairperson: Dr. Aman Shakya
Assistant Professor,
Department of Electronics and Computer Engineering,
Institute of Engineering, Tribhuvan University.

Date: November, 2019

DEPARTMENTAL ACCEPTANCE

The thesis entitled **"User Behavior Modeling and Anomaly Detection in Cybersecurity Data Using Deep Learning",** submitted by **Balaram Sharma** in partial fulfillment of the requirement for the award of the degree of **"Master of Science in Computer System and Knowledge Engineering"** has been accepted as a bonafide record of work independently carried out by him in the department.

.....

Assoc. Prof. Dr. Surendra Shrestha

Head of the Department

Department of Electronics and Computer Engineering,

Pulchowk Campus,

Institute of Engineering,

Tribhuvan University,

Nepal.

ACKNOWLEDGEMENT

My profound gratitude and deepest appreciation goes to my Thesis Supervisor **Assist. Prof. Dr. Basanta Joshi,** Pulchowk Campus, for his consistent support, guidance and suggestion at various stages of this thesis work.

I would like to express my special thanks of gratitude to the Department of Electronics and Computer Engineering (DOECE) and to our Head of Department **Assoc. Prof. Dr. Surendra Shrestha** for providing us with the golden opportunity to explore our interest and ideas in the field of engineering through this thesis. I would like to provide my sincere gratitude to our MSCSKE Coordinator **Assist. Prof. Dr. Aman Shakya** for providing us with necessary details and ideas throughout out thesis work. I would also like to thank **Prof. Dr. Shashidhar Ram Joshi**, **Prof. Dr. Subarna Shakya**, **Assoc. Prof. Dr. Sanjeeb Prasad Pandey** for their constant support, comments and encouragement during this research activity.

Finally, I would like to thank all our teachers and friends who have helped me directly or indirectly for encouraging me with this thesis topic and research decision.

Balaram Sharma
073/MSCS/653

ABSTRACT

User behavior analytics is one of the trending topics nowadays in the field of cybersecurity. Traditionally people were not much concerned about attacks originating from intentional/unintentional actions of employees within the organization. The daily news about data breaches of different organizations from their own employees, the employers are becoming more concerned about the necessity to monitor user's behavior within the network. This thesis work proposes an approach for user behavior analytics. In this thesis work, a mechanism to process and analyze raw events related to user actions have been described. The CERT insider threat dataset has been used for the research work. For each user in the dataset, the feature vectors for machine learning are prepared by extracting key information from corresponding raw events and aggregating the frequency of actions within the session window. The unsupervised learning called LSTM Autoencoder has been implemented for behavior learning and anomaly detection. The whole dataset i.e. feature vectors are divided chronologically with time ordering into training, validation and testing sets. The model is taught to learn normal behavior. During the testing phase, when the unseen behavior or anomaly pattern is fed, the model produces high reconstruction error which is an indication of an anomaly. From the experiment, it was found that test accuracy of 89.74%, True Positives of 90.53% and False Positives of 10.26%.

Keywords*: User Behavior Analytics, Anomaly Detection, LSTM RNN Autoencoder.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AD	Active Directory
APT	Advanced Persistent Threats
AUC	Area Under the Curve
AWS	Amazon Web Services
C&C	Command and Control
CERT	Computer Emergency Response Team
DARPA	Defense Advanced Research Projects Agency
DBN	Deep Belief Network
DNN	Deep Neural Network
FP	False Positives
FPR	False Positive Rate
IAM	Identity Access and Management
ICT	Information and Communication Technology
k-NN	k-Nearest Neighbors
LSTM	Long-Short Term Memory
ML	Machine Learning
MSE	Mean Squared Error
NSA	National Security Agency
OCSVM	One Class Support Vector Machine
PCA	Principle Component Analysis
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SIEM	Security Information and Event Management
TN	True Negative
TNR	True Negative Rate
TP	True Positives
TPR	True Positive Rate
UBA	User Behavior Analytics
UEBA	User and Entity Behavior Analytics
VPC	Virtual Private Cloud
VPN	Virtual Private Network

CHAPTER 1: INTRODUCTION

1.1 Introduction

Nowadays cybersecurity is becoming a subject of public concern. With a rapidly growing network and technologies, our life is being easier in every aspect. However, it also poses a challenge to protect the information from a potential insider or outsider threats. We are becoming heavily dependent on Information and Communication Technology (ICT) across all aspects of our cyber-physical society. As a result, the need for cybersecurity is becoming increasingly important. Since information is the main asset, cybersecurity is essential for all of us; individuals or organizations.

Basically, there are two types of security threats for an organization; outsiders and insiders. The main intent of this research work is to detect insider threats by analyzing the activity data of users within the organization. The LSTM RNN model is built on the normal (non-anomalous) data of the users i.e. normal profiles are created. When user behaves differently model produces high error to recognize the data, which can be the indication of an anomaly.

The 2018 Insider Threat Report reveals that insider threats are increasing day by day due to malicious/deliberate or accidental/unintentional actions of users within the organization [1]. Due to this organizations have to bear information leakage and financial loss. Insider threats are being complex since attackers might behave like normal peoples and is a growing challenge for employers. An insider attack is generally defined as any actions taken by an employee that are potentially harmful to the organization; e.g. unsanctioned data transfer or sabotage of resources or unauthorized access of resources. Insider threat may manifest in various and novel forms motivated by differing goals, ranging from a disgruntled employee subverting the prestige of an employer to advanced persistent threats (APT), orchestrated multi-year campaigns to access and retrieve intelligence data [2].

These increasing volume of insider threats are pressing the cybersecurity professionals to take more action and deploy User Behavior Analytics (UBA) tools and solutions to help detect, classify and alert anomalous behavior. According to the Gartner, User behavior analytics is “a cybersecurity process about the detection of insider threats, targeted attacks,

and financial fraud. UBA solutions look at patterns of human behavior and then apply algorithms and statistical analysis to detect meaningful anomalies from those patterns anomalies that indicate potential threats. Instead of tracking devices or security events, UBA tracks a system's users" [3]. To detect anomalous behavior of a user within a network, we need to collect his/her activity data over a period of time and analyze the pattern.

1.2 Problem Statement

Though the active attacks e.g. a sudden brute force attack, can be detected by modern firewalls, antivirus software, intrusion detection systems etc., what user does within the network passively are mostly out of their monitoring scope. Mostly the attackers perform certain malicious actions sitting under the rule. For example, within an organization a user gets the authentication credentials of privileged accounts and steals valuable information. To detect activities like this, we need to monitor behavior of users over a period of time. Many data and security breaches has been done by the users within the organizations. These things signify necessity to monitor User Behavior within a network.

Several research works done in this domain have implemented different algorithms with statistical method for prediction. However, each user has some pattern of performing activities. The proposed research work looks into sequence of events or behavioral patterns for detecting abnormal behavior. Since model needs to remember past states as well, LSTM RNN has been proposed. While implementing ML algorithms, we need to prepare feature vector that is input to the algorithm. Taking fixed time window for feature vector preparation has potential to miss anomalous pattern. In short time window, anomalous pattern might be split. Long time window, becomes too much generic. One solution could be instead of taking fixed time window, we can take user's session.

1.3 Objectives

- Extraction of features from raw events of user activity and preparing feature vectors for ML.
- Apply Unsupervised Deep Learning (LSTM RNN Autoencoder) algorithm for modeling behavioral pattern of user's activities and anomaly detection.

CHAPTER 2: LITERATURE REVIEW

2.1 Need for Security

In the era of the Internet, protecting your organization's information has become just as important as guarding your property. Information is considered as lifeblood of a successful and profitable business, and employees of the organization work as veins and arteries to pass this information through. No matter how large or small your company is, you need to have a plan to ensure the security of your information assets. For an organization, information is valuable and should be appropriately protected. Security is to combine systems, operations, and internal controls to ensure integrity, confidentiality and availability of data and operation procedures in an organization. Information is main asset for all the organization nowadays and protecting information from outside and inside threats and attacks is absolutely necessary though it is challenging job. In recent years the major network and data breaches have become larger in scale and becoming more serious ranging from theft of corporate data to targeted campaigns against governments. The need to identify network security threats has even been greater [4].

2.2 User Behavior Analytics

According to biologists, "behavior is the internally coordinated responses of whole living organisms to internal and/or external stimuli" – so basically behavior is everything that we are doing consciously. Similarly, digital behavior is everything that we are doing in the digital world. Typing characteristics, screen resolution of our computer, smartphone or tablet, favorite applications or websites and many others are our digital footprints that typifies us so strongly than our habits. As parents are able to recognize and differentiate their children based on the sound of their footsteps, User Behavior Analytics solutions are able to recognize and differentiate users based on their digital activities [5]. When user exhibits unusual behavior or activities from his/her regular behavior then the behavior is called as abnormal behavior. For instance, an employee(User) does large volume of file copy from an organization's network to external sources like dropbox, google drive etc., however such drives are actually less frequently used for an organization. So, this behavior can an anomalous behavior.

The user's behaviors are interlinked with security of an organization. The major data breaches have been happening due to the insider users. Also, the Advanced Persistent Threats(APT) are targeted to the user of an organization. We could take an example of Edward Snowden; who copied and leaked classified information from the National Security Agency (NSA) in 2013 without authorization [6].

There has been much research in Anomalous User Behavior Detection. Many SIEM vendors have already implemented the User and Entity Behavior Analytics solutions. There are more research works as well in this field.

2.3 Related Works

Aaron Tuor, Samuel Kaplan and et.al [2] Present an online unsupervised deep learning system to filter system log data for analyst review. Because insider threat behavior is widely varying, they do not attempt to explicitly model threat behavior. Instead, novel variants of Deep Neural Networks (DNNs) and Recurrent Neural Networks (RNNs) are trained to recognize activity that is characteristic of each user on a network and concurrently assess whether user behavior is normal or anomalous, all in real time. To aid analysts in interpreting system decisions, their model decomposes anomaly scores into a human readable summary of the major factors contributing to the detected anomaly (e.g. that the user copied an abnormally large number of files to removable media between 12am and 6am).

Derek Lin, Baoming Tang and Qiaona/Joanna Hu [4] have implemented principal component analysis (PCA) algorithm for Behavior Learning and Analysis on the Enterprise Event Logs. They have collected data from different security products and services like: Active Directory Services, VPN, Web Proxy Products, Identity management products, Cloud Services etc. The logs messages from different sources are first parsed and normalized to meta events to reduce data complexity. User's daily traffic is represented by an array of frequency counts of meta events.

Xiangyu Xi, Tong Zhang and et.*al* [7], the authors present an overview of User Behavior Analytics Platform built to collect logs, extract features and detect anomalous users which may contain potential insider threats. They have used multi-algorithms ensemble approach;

combining OCSVM, RNN and Isolation Forest for the experiment. The experiment shows that the system they built with an ensemble of unsupervised anomaly detection algorithms can effectively detect abnormal user behavior patterns.

Fanzhi Meng, Fang Lou, Zhihong Tian and et.*al* [8] proposed a novel attribute classification insider threat detection method based on long short-term memory recurrent neural networks (LSTM-RNNs). To achieve high detection rate, they have integrated several components. They have used CERT insider threat dataset v6.2 and threat detection recall as their performance metric. Their experimental results that RNN method has good performance as compared to other threat detection methods like; k-Nearest Neighbor, Isolation Forest, Support Vector Machine etc.

Madhu Shashanka, Min-Yi Shen and Jisheng Wang [9] present the implementation of UEBA solution for Niara Security Analytics Platform. They have used SVD-based algorithms to detect anomalies of interest from real-world dataset of network traffic collected within the Niara internal network over a span of a 3 months. The entire dataset collected comprises 1.3 Billion raw data records where each record corresponds to a network layer-4 conversation. They have used data sources such as network packets and logs to identify anomalous behavior of users, IP addresses and devices within an enterprise network.

Insider threats are a considerable problem within cyber security and it is often difficult to detect these threats using signature detection. Owen Lo, William J. Buchanan and et. al [10] applied Distance Measurement Methods for Insider Threat Detection. Their work builds on a published method of detecting insider threats and applies Hidden Markov method on a CERT data set (CERT r4.2) and they analyze a number of distance vector methods (Damerau–Levenshtein Distance, Cosine Distance, and Jaccard Distance) in order to detect changes of behavior, which are shown to have success in determining different insider threats.

Tabish Rashid, Ioannis Agrafiotis and Jason R. C. Nurse [11], the authors investigated the task of detecting insider threats through a novel method of modelling a user's normal behaviour in order to detect anomalies in that behavior which may be indicative of an attack. Specifically, they use Hidden Markov Models to learn what constitutes normal behavior, and

then use them to detect significant deviations from that behavior. Their result shows that this approach is indeed successful at detecting insider threats, and in particular is able to accurately learn a user's behavior.

2.4 Datasets

2.4.1 Synthesized dataset

For the research work, synthesized dataset has been in common use. One of the simulated dataset is CERT Insider Threat dataset [12]; developed in partnership with ExactData, LLC, and under sponsorship from DARPA I2O. It is a collection of synthetic insider threat test datasets that provide both background and malicious actor synthetic data.

2.4.2 Real Data

The SIEM vendors use real data collected from various sources within the network [4]. Also, there are many research works done in real data. Some of the data sources and their use in User Behavior Analytics are listed below:

- **Packets**

  Deep Packet Inspection (DPI) can be used to extract L4-L7 metadata attributes that provide a granular view of the user's activity.

- **Netflow**

  Many environments are already instrumented to provide this data (e.g. Netflow from Cisco switches, VPC Flow from AWS) that can be used to detect lateral spread and exfiltration attempts.

- **DNS**

  DNS are a rich source of information to track C&C and exfiltration activities.

- **SaaS Logs**

Can be accessed through enterprise accounts at cloud providers like Box and Office365 to track user access to these enterprise assets.

- **Firewall, Web Proxy Logs, etc.**

They are an alternate source of activity information either at the ingress/egress of an organization's perimeter, or in front of a datacenter where they can be used to track access to high value assets.

- **Windows EventLogs**

Many analytics can be done from window's eventlogs such as authentication attempts, lateral movement, malicious software installation etc.

2.5 Review of Algorithms commonly used in UBA

Machine learning is a core capability in the product category that Gartner calls User and Entity Behavioral Analytics (UEBA). Enterprise security teams are now turning to UEBA tools for a new dimension of attack detection and threat hunting. Unlike rule-based systems, UEBA features machine learning and other analytics techniques to automatically find the threats that evade existing security tools [13]. Different Machine learning algorithms have been used for User Behavior Analytics. Some of the ML algorithms commonly used in user behavior analytics are described below:

2.5.1 Neural Networks

Neural Networks are commonly used in UBA researches.

1. **Replicator Neural Network**

For unsupervised learning, Replicator Neural Network is popular. The paper [8] uses Replicator Neural Network for anomalous user behavior detection. It is an artificial feed-forward multi-layer neural network with an output layer having the same number of nodes as the input layer. The purpose of Replicator Neural Network is to produce the output data which as is similar as the input data. Replicator Neural Network is effective in anomaly

detection as an unsupervised machine learning algorithm because anomalies are few and there exist some common patterns in normal data. By the trained RNN, the common patterns representing bulk of the data can be well reproduced, while anomalies will have a much higher reconstruction error [7].

The reconstruction error for a d-dimensional instance $x = (x_1, x_2, ..., x_d)$ is computed as follow:

$$e = \sum_{i=1}^{d} (x_i - y_i)^2 \tag{2.1}$$

in which d is the dimension of input vector x and $y = (y_1, y_2, ..., y_d)$ is the reconstructed output.

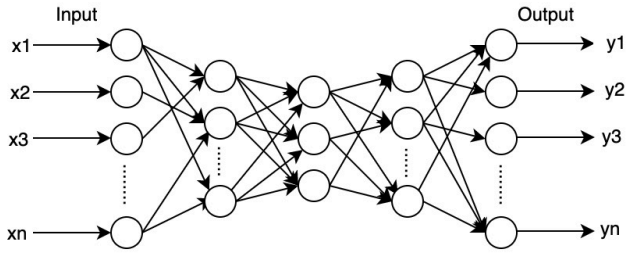


Figure 1. Replicator Neural Network

2.5.2 One Class SVM

OCSVM, proposed by Scholkopf [14], has been applied to anomaly detection. Figure 2. shows the OCSVM algorithm maps input data into a high dimensional feature space via a kernel and iteratively finds the maximal margin hyperplane which best separates the training data from the origin.

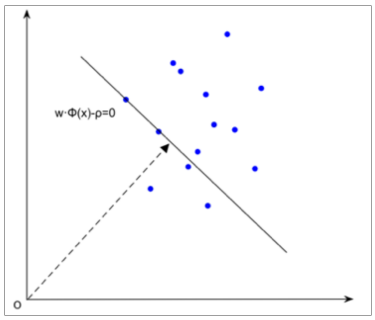


Figure 2. Diagram of OCSVM hyperplane

2.5.3 Principle Component Analysis (PCA)

Principal component analysis is a tool for finding patterns in high-dimensional data. For a set of m users and n dimensions, data can be arranged as m×n matrix X, whose rows correspond to users and whose columns correspond to user behavior features discussed above. PCA then extracts common patterns from the rows of X in an optimal manner. These common patterns are called principal components, and their optimality property is as follows: over the set of all unit vectors having n elements, the first principal component is the one that captures the maximum variation contained in the rows of X. More formally, the first principal component v1 is given by [15]:

$$v1 = \arg \max_{||v||=1} ||Xv|| \quad (2.2)$$

The expression Xv yields the inner product (here, equivalent to the correlation) of v with each row of X; so v1 maximizes the sum of the squared correlations.

2.5.4 Deep Learning

The popularity and capability of deep learning algorithms are not only limited in image processing, text processing, audio and video processing but also has been increasing for temporal data analysis and anomaly detection. Paper [2] uses RNN and DNN for unsupervised learning. Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence [16]. Learning can be supervised, semi-supervised or unsupervised. Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain.

1. Recurrent Neural Network

RNN; a variant of artificial neural network, is a class of supervised machine learning method made of artificial neurons with one or more feedback loops. Actually, the feedback loops are recurrent cycles over time or sequence [17]. The benefit of Recurrent connections is that the RNN model can improve performance by leveraging

their ability to understand sequential dependencies. However, sometimes, the memory produced from the recurrent connections can severely be limited by the algorithms employed for training RNNs. As a result, the model might be victim to exploding or vanishing gradients during the training phase, resulting in the network failing to learn long term sequential dependencies in data [17].

To mitigate the exploding or vanishing gradients problem, the long-short term memory (LSTM) RNNs has been designed.

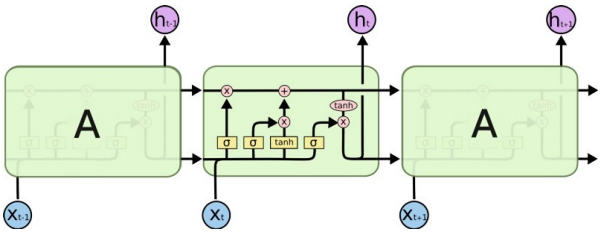


Figure 3. LSTM Memory Cell

RNN model is organized in cells which include several operations. LSTM has an internal state variable, which is passed from one cell to another and modified by **Operation Gates** [18]. The information flow in LSTM networks is controlled by three gates (f, i, o) [8]:

Forget Gate

$$f_t = \sigma\left(W_f . [h_{t-1}, x_t] + b_f\right) \tag{2.3}$$

It is a sigmoid layer that takes the output at *t-1* and the current input at time *t* and concatenates them into a single tensor and applies a linear transformation followed by a sigmoid. Because of the sigmoid, the output of this gate is between 0 and 1. This number is multiplied with the internal state and that is why the gate is called a forget gate. If *ft=0* then the previous internal state is completely forgotten, while if *ft=1* it will be passed through unaltered.

Input Gate

$$i_t = \sigma\left(W_i . [h_{t-1}, x_t] + b_i\right) \tag{2.4}$$

The input gate takes the previous output and the new input and passes them through another sigmoid layer. This gate returns a value between 0 and 1. The value of the input gate is multiplied with the output of the candidate layer.

$$C_t = \tanh(W_C.[h_{t-1}, x_t] + b_C) \quad (2.5)$$

This layer applies a hyperbolic tangent to the mix of input and previous output, returning a candidate vector to be added to the internal state.

The internal state is updated with this rule:

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (2.6)$$

The previous state is multiplied by the forget gate and then added to the fraction of the new candidate allowed by the output gate.

Output Gate

$$O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (2.6)$$

$$h_t = O_t * \tanh C_t \quad (2.7)$$

It controls how much of the internal state is passed to the output and it works in a similar way to the other gates.

In most of the real-world problems, it is hard to find the labelled data for supervised learning. In that case deep learning models can also be trained in unsupervised way. In case of anomaly detection problem, the anomalies are generally very rare as compared to normal records. In such case as well directly applying supervised learning, might not perform well because of class imbalance. The unsupervised methods such as Autoencoders can be used for such cases. Autoencoders consist of one input layer, one or more hidden layer and one output layer. Generally, they have less units in their hidden layers compared to the input or output layers. The units in inputs and outputs layers of autoencoder is always same. They can be used for as unsupervised for anomaly detection problem. An autoencoder has two phases which are encoding and decoding. In encoding, it takes the given input and tries to express the input with lesser units than the input units by using its hidden layer(s). In decoding layer, an autoencoder tries to reconstruct the given input with using the encoded information in its hidden layer(s) [19].

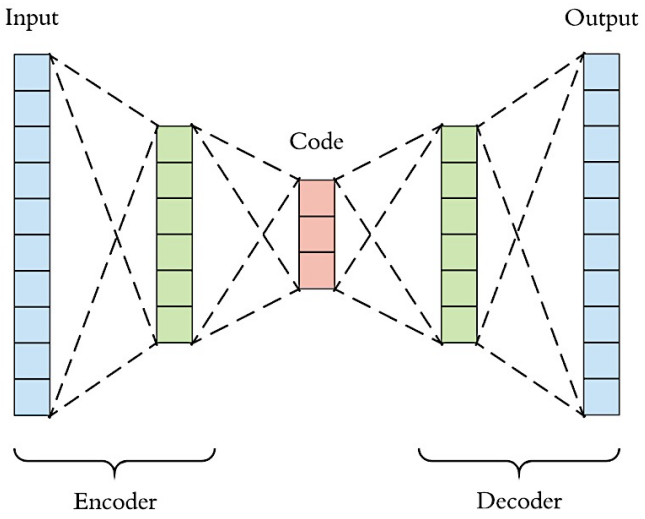


Figure 4. Basic Autoencoder

The LSTM has 3D input shape. The input vector comprises of samples, time steps, and features with the shape: num_samples, num_timesteps, and num_features respectively.

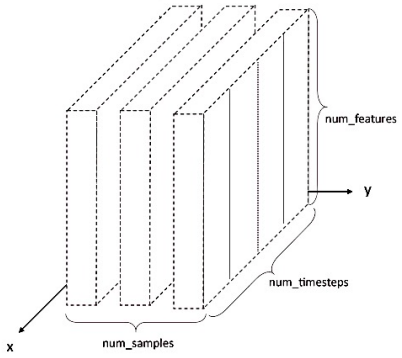


Figure 5. Input shape for LSTM RNN

The num_samples are the number of data rows. The num_timesteps are the past observation for the feature, i.e., a lag variable. The num_features are the total attributes or columns [20].

CHAPTER 3: METHODOLOGY

3.1 Data Collection

The user behavior modeling and evaluation is performed in the CERT insider threat dataset. The CERT Division, in partnership with ExactData, LLC, and under sponsorship from DARPA I2O, has generated a collection of synthetic insider threat test datasets. These datasets provide both synthetic background data and data from synthetic malicious actors [21]. Basically, there are five types of raw activity data or events inside the dataset. They are listed below with some sample events:

1. **device**: contains events related to device access

```
{J1S3-L9UU75BQ-7790ATPL},01/02/2010 07:21:06,MOH0273,PC-6699,Connect
{N7B5-Y7BB27SI-2946PUJK},01/02/2010 07:37:41,MOH0273,PC-6699,Disconnect
```

2. **http**: contains url visited by each employee

```
{V1Y4-S2IR20QU-6154HFXJ},01/02/2010 06:55:16,LRR0148,PC-4275,http://msn.com/The_Human_Centipede_First_Seq
uence/katsuro/arjf309875127.htm,remain representatives consensus concert although connect component colla
borated mind sea 200 decision mixture regarding days warners rich fifth 1991 return 1988 apologized unite
d crew tension particular allied rhythm making japanese enhance pop myself whole allmusic bras at focused
 throughout another touches want references
{Q5R1-T3EF87UE-2395RWZS},01/02/2010 07:00:13,NGF0157,PC-6056,http://urbanspoon.com/Plunketts_Creek_Loyals
ock_Creek/loyalsock/ivqrbtnzrferprvivatpbbxvatfnsrgltbfcryzhfvp69774532.jsp,festival off northwards than
congestion partnership married 1830 acquired gentleman closely 17 take 1841 hundreds indicated allowing o
rder characters disjunct hub temperature examples expanded adapted tribute
```

3. **logon**: has records the logon and logoff activities by each employee

```
{H3B1-S1YW87QV-8953UWJM},01/02/2010 09:55:53,NGF0157,PC-6056,Logon
{A3I7-T5FC59LD-3522VDHH},01/02/2010 17:05:00,NGF0157,PC-6056,Logoff
```

4. **email**: contains information about email transactions between employee

```
{H7P4-X6NX60ZQ-0801PKMY},01/02/2010 10:25:56,ALB0892,PC-2744,Stacey_D_Solomon@juno.com,,,ALB5@cox.net,192
85,0,begins ci point liquid front 9 pm frontal cuba course as long providing significance wide prediction
 begins random mbar greenhouse cycle taken signal various traffic found ground 28 contrast push formed ni
mbostratus random put close 24 fanning optical such present 30 particular james wide cumulonimbus visible
 century not heavier enters larger
{P0X3-X1IF35FW-9915SANA},01/02/2010 10:25:59,ALB0892,PC-2744,Blythe.Veda.Cooke@dtaa.com,,,Arthur.Lucian.B
onner@dtaa.com,24921,2,2003 very frenzy foreign a know smith endorsement reprised ryder themselves affect
ed fleshy proceeds title abandoned non win police notable france described he greatest rival forgotten ra
ising cinematic twenties san undergo intently charitable probably nomination reinforced protection do 151
 couples obtained punk spots they impostor gestures funeral united drugs wallace seven gamble way achieve
d twice teacher fulfilling ended
```

5. **file:** contains information about file access made by employees.

```
{S8N7-Y1KE53NB-3318TRZC},01/02/2010 14:49:42,HPH0075,PC-2417,CIRZIA7I.doc,D0-CF-11-E0-A1-B1-1A-E1 measuri
ng national even theories tried appropriately frames multiple pears transported routes soon rip processin
g 1980s protet 1938 historic launches habitat exist since reconstituted museum announced saturday migrati
on lake pebbles legislation olive 20th ships
{C9I3-L3LN67UM-4413TNBV},01/02/2010 15:07:09,HSB0196,PC-8001,FKXP0CGT.doc,D0-CF-11-E0-A1-B1-1A-E1 treaty
proceed substance asked meaning during vladas formed meaning commonwealth en ties seeking approved amongs
t powers 1941 up en society led stulginskis whose road existing nearly periodicals freeholders major olde
r spanning contents bearer l with thought macdonald subject corner fashion 94 arnold holden safety 1974 m
armot severe towards fearing agenda third forming
```

Following is the statistics of data in CERT v4.2 dataset.

Table 1. Data Statistics

Event Type	**Count**
Device	405,381
Email	2,629,980
File	445,582
HTTP	28,434,424
Logon	854,860
Total Events	**32,770,227**
Total Threat Events	**7,323**
No of Users	**1000**
No of Days	**502**

3.2 Threat Scenarios

The CERT dataset consists users and their daily activities within an organization over the period of almost one and half year. As in real where the threat events are usually rare, the dataset also has very few threat events. There are three different use cases in the CERT v4.2 dataset [21]. If the model recognizes the sequence of such activities, the anomaly will be detected.

The insider threat cases are as follows:

Table 2. Insider threat scenarios

Threat Scenarios	Description
1	User who did not previously use removable drives or who has not worked after hour, now begins logging in after hour. He/she use removable drives and upload the files to the hacking or cloudstorage such as wikileaks.org. The user leaves the organization thereafter.
2	The user begins surfing job websites and soliciting employment from a competitor. Such activities were not performed earlier. Before leaving the company, they use a thumb drive (at remarkably higher rates than their previous activity) to steal data.
3	System administrator (who has access to multiple resources/computers) becomes disgruntled. He/she downloads a keylogger and uses a thumb drive to transfer it to his supervisor's machine. The next day, he uses the collected keylogs to log in as his supervisor and send out an alarming mass email, causing panic in the organization. He/she leaves the organization immediately.

3.3 Architecture for user behavior modeling and anomaly detection

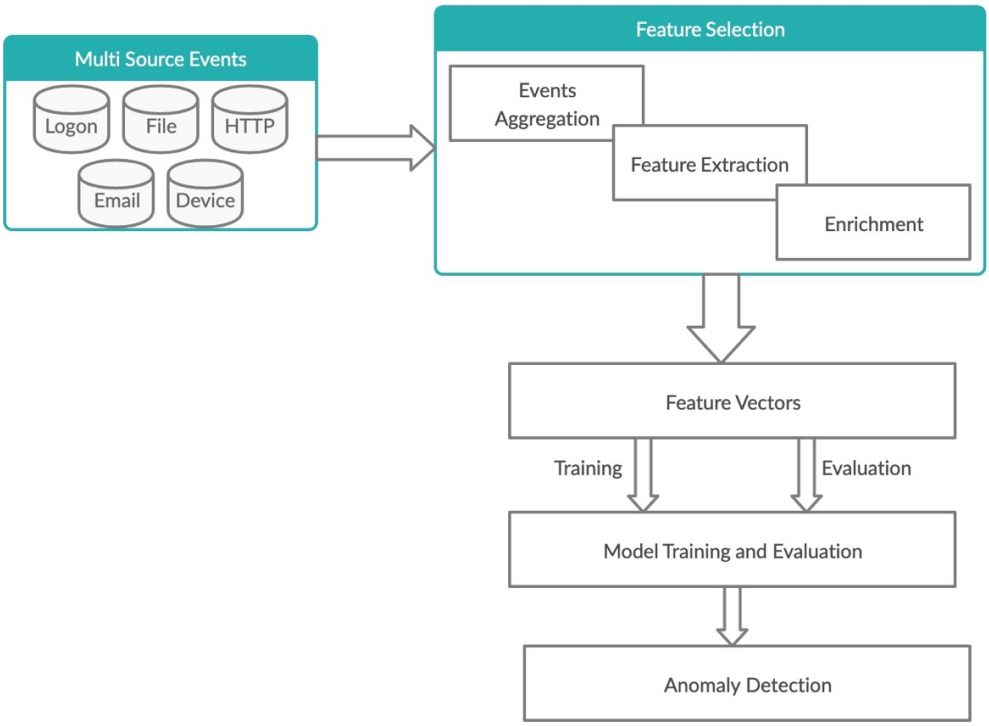


Figure 6. Architecture for user behavior modeling and anomaly detection

The CERT Dataset contains 5 log files (each log file contains raw events) which describe a particular kind of activity for all users. Over the course of almost 500 days, 1000 users generate 32,770,227 events (log lines). Among these there some events manually injected by domain experts, representing three insider threat scenarios taking place. Additionally, user attribute metadata, such as role, project, functional unit, department, team and supervisor, is included [8]. To train and test the method, the dataset will be chronologically divided into three subsets: training, validation and testing. The training subset will be used for model selection and hyper-parameter tuning, while the testing subset will be held out for assessing generalization performance.

3.4 Feature Selection

The data provided are just the raw events or activities. The major challenge of anomaly detection is the feature engineering, i.e. how to extract features and use them for the ML model so that model can learn the behavior of each user. Most of the papers (e.g. [2], [4], [7],

[22]) has taken the fixed time window and aggregate each user's activities occurring within the time window to prepare the feature vector. Taking fixed time window for feature vector preparation has sometimes potential to miss anomalous pattern. In short time window, anomalous pattern might be split. Long time window, becomes too much generic. One solution could be instead of taking fixed time window, we can take user's session.

3.4.1 Event Aggregation

The Event aggregator combines the events come from different sources, such as: http events, file events, email events, logon events and device events into a common data format in addition to time-ordering them.

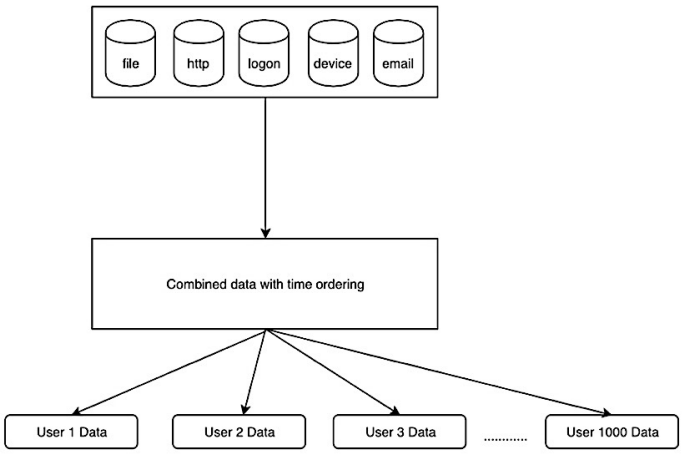


Figure 7. Event Aggregation

3.4.2 Feature Extraction

1. **Session Calculation**

In this research session based feature selection is performed. The user session is defined as the activities performed by the user within logon--logoff duration. So, the session duration is not fixed i.e. variable. In the dataset information, it has been mentioned that, the logon activity precedes the other activities.

The user session is defined as time interval between, login to his/her assigned pc and logoff to the same pc. Three things are extracted as a user session from Logon Events:

1. **Start_ts:** Logon time in assigned pc

2. **End_ts:** Logoff time
3. **User:** The corresponding user.

Once each user's sessions are extracted, the other events belonging in between the start_ts and end_ts of the corresponding user are included within that window. From there features are calculated. For example, the session for user NGF0157 is as below:

```
01/02/2010 06:49:00,NGF0157,PC-6056,Logon
01/02/2010 17:05:00,NGF0157,PC-6056,Logoff
```

Figure 8. Example of a session

```
1/2/10 6:49,NGF0157,PC-6056,Logon
01/02/2010 07:00:13,NGF0157,PC-6056,http://urbanspoon.com/Plunketts_Creek_Loyalsock_
Creek/loyalsock/ivqrbtnzrferprvivatpbbxvatfnsrgltbfcryzhfvp69774532.jsp,festival off
 northwards than congestion partnership married 1830 acquired gentleman closely 17 t
ake 1841 hundreds indicated allowing order characters disjunct hub temperature examp
les expanded adapted tribute
1/2/10 7:56,NGF0157,PC-6056,Connect
1/2/10 8:53,NGF0157,PC-6056,ZYPY973Y.pdf,25-50-44-46-2D it 28 no settle hundred ther
e tell appearance son leura sales for emergence advised promoted being medical howev
er regarded martin eight age 1931 promoted comparing 73 graceful played evatt top se
ries 1909 news hunt prevented likely outside however no
1/2/10 9:06,NGF0157,PC-6056,Disconnect
01/02/2010 11:19:52,NGF0157,PC-6056,Libby.Rosalyn.Richard@dtaa.com;Bethany.Xerxes.Jo
hnson@dtaa.com,Sasha.Rina.Huffman@dtaa.com,Nissim.Gil.French@dtaa.com,Nissim.Gil.Fre
nch@dtaa.com,28681,1,robert defeat controversy passing episodes meets batarang tools
 tools shines cloud child riddler killer 900 lord sense analysis relaxed boomerang
1/2/10 17:05,NGF0157,PC-6056,Logoff
```

Figure 9. Example of activities belonging to a session

2. Feature Extraction

Since data provided is raw events, so there is not exact rule about how many features to construct from each log sources. The no. of features taken varies in different research works. Feature selection also depends on the use case. There can be two types of features; Numerical and Categorical user attributes. In this research work, numerical selection is done according to the information provided in *readme.txt* file of dataset and by taking reference of some the research works: [2] [22], [23] and [24]. The numerical features are calculated by aggregating the occurrence of each events within the session.

The table below shows the features extracted from each event source:

Table 3. Numerical features

Domain	**Features**
Session (logon-logoff)	start_ts, end_ts, user
Logon activity	#logon_on_own_pc_normal, #logon_on_other_pc_normal, #logon_on_own_pc_off_hour, #logon_on_other_pc_off_hour, #logon_hour, #day_of_a_week
Device activity	#device_connects_on_own_pc_normal_hour, #device_connects_on_other_pc_normal_hour, #device_connects_on_own_pc_off_hour, #device_connects_on_other_pc_off_hour
File activity	#documents_copy_own_pc, #documents_copy_other_pc, #exe_files_copy_own_pc, #exe_files_copy_other_pc, #documents_copy_own_pc_off_hour, #documents_copy_other_pc_off_hour, #exe_files_copy_own_pc_off_hour, #exe_files_copy_other_pc_off_hour
HTTP activity	#neutral_sites, #job_search, #hacking_sites, #neutral_sites_off_hour, #job_search_off_hour, #hacking_sites_off_hour
Email activity	#total_emails, #int_to_int_mails, #int_to_out_mails, #out_to_int_mails, #out_to_out_mails, #internal_recipients, #external_recipients, #distinct_bcc, #mails_with_attachments, #after_hour_mails

Table 4. Categorical features

Categorical Features	
User Roles	**Psychometric**
User	O
Role	C
Functional Unit	E
Department	A
business unit	N

3.4.3 Enrichment

When features are extracted for each user, each user is enriched by user attribute metadata e.g. user role, project, functional unit, department etc. This information is also important while modeling user's behavior since employee's behavior also depends on the roles and responsibilities.

3.4.4 Feature vector generation

After feature selection, for each user u, for each session time t, the feature vector is:

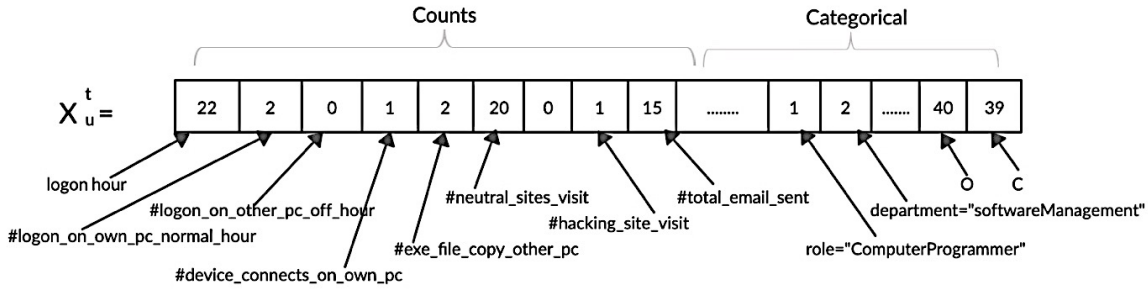


Figure 10. An example of feature vector

3.4.5. Data Normalization

The data for each user for machine learning are normalized to a range [0,1] using min max scaling.

For each feature x_i for each user in the dataset, the features are normalized as:

$$x_i = \frac{(xi-(xi)\min)}{(xi-(xi)\max)} \quad (3.1)$$

3.5 Unsupervised LSTM RNN Model

LSTM RNN model has been widely used for temporal data analysis because they have the capability to understand sequential dependencies. In the behavioral analytics as well, there exists pattern of activities of a user or employee within an organization.

Since we have very few anomaly data in the dataset, we don't use anomaly data in the training and validation phase. The model is trained on normal sequences i.e. model learns the normal behavior of the users. During the training phase, input is reconstructed in the output, so reconstruction error is minimized. In the testing phase, the positive as well as negative scenarios are fed to the network. For the unseen anomaly data, the network should produce high reconstruction error.

A user usually has multiple attributes; and RNN model will be constructed for each user. We have input sequence $\{X_t\}^u$, where $u = \{u_1, u_2, \ldots\ldots\ldots, u_n\}$, are total users. At time t, for a user u, the input sequence is: $X_t = \{x^t_1, x^t_2, x^t_3, \ldots\ldots\ldots\ldots, x^t_n\}$. Where n is the number of features in the dataset.

For each X_t, the framework for the generic RNN for the i^{th} column of X_t is given as follows [25]:

$$h_{t_i} = f(W\, x_{t_i} + R h_{(t-1)_i}) \tag{3.2}$$

Where h_{t_i} is the state vector and x_{t_i} is the input vector. The f is relu function which is defined as:

$$f(x) = \max(0, x) \tag{3.3}$$

The LSTM network implemented is autoencoder model which has encoder and decoder section. The output of encoder block for each Xt is defined as:

$$h_{t_i} = f f^{en}(x_{t_i}, h_{(t-1)_i}) \tag{3.4}$$

Now the h_{t_i} is the input to the decoder block of the network. The Output of the decoder is defined as:

$$\hat{h}_{t_i} = f^{dec}(h_{t_i}, \hat{h}_{(t-1)i}) \tag{3.5}$$

Now the reconstructed output of the corresponding input x_t is:

$$\hat{x}_{t_i} = f(\hat{h}_{ti}) \tag{3.5}$$

After the reconstructed input is retrieved, the mean squared error is calculated to update encoder-decoder parameters of the LSTM model. The formula for the mean squared error is:

$$\text{MSE} = \sum_{i=1}^{ni} |\hat{x}_{ti} - x_{ti}|^2 \tag{3.6}$$

In Long Short-Term Memory (LSTM) RNN architecture (Hochreiter and Schmidhuber 1997), the hidden state h_t at time t is a function of a long-term memory cell, c_t . The output of hidden depends on the input sequence and cell states as follows [2] [25]:

$$\tilde{\boldsymbol{c}}_t = g\left(\boldsymbol{W}^{(\tilde{c})}\boldsymbol{x}_t + \boldsymbol{R}^{(\tilde{c})}\boldsymbol{h}_{t-1} + \boldsymbol{b}^{(\tilde{c})}\right) \tag{3.7}$$

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}^{(i)}\boldsymbol{x}_t + \boldsymbol{R}^{(i)}\boldsymbol{h}_{t-1} + \boldsymbol{b}^{(i)}\right) \tag{3.8}$$

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}^{(f)}\boldsymbol{x}_t + \boldsymbol{R}^{(f)}\boldsymbol{h}_{t-1} + \boldsymbol{b}^{(f)}\right) \tag{3.9}$$

$$\boldsymbol{c}_t = \boldsymbol{D}_t^{(i)}\tilde{\boldsymbol{c}}_t + \boldsymbol{D}_t^{(f)}\boldsymbol{c}_{t-1} \tag{3.10}$$

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}^{(o)}\boldsymbol{x}_t + \boldsymbol{R}^{(o)}\boldsymbol{h}_{t-1} + \boldsymbol{b}^{(o)}\right) \tag{3.11}$$

$$\boldsymbol{h}_t = \boldsymbol{D}_t^{(o)} l(\boldsymbol{c}_t), \tag{3.12}$$

3.6 Anomaly Detection

The LSTM network is trained in the normal data, so the model learns normal behavior of the users. In the testing phase both normal as well as anomaly sequences are fed to the network. Since, the network has learned normal pattern, it should produce low reconstruction error in normal data and high reconstruction error in abnormal patterns. The reconstruction error is calculated for each input sequence of the test data. A threshold θ, is defined to separate normal and anomaly sequence. The threshold value θ is defined as,

$$\theta = \beta * mean(e_0, e_1, e_2 \ldots e_n) \tag{3.13}$$

Where β is some constant and e_i is the reconstruction error for i^{th} sequence.

The predicted class is defined as below:

$$\text{Predicted Class} = \begin{pmatrix} \text{Anomaly} & if\ e > \theta \\ Normal & else \end{pmatrix} \quad (3.14)$$

3.7 Evaluation

3.7.1 Confusion Matrix

The confusion matrix is used for the validation and performance evaluation of Model. It gives matrix as output which describes complete performance of the model.

Table 5. Confusion matrix for anomaly detection

		Predicted Class	
		Normal(Negative)	Anomaly(Positive)
Actual Class	Normal(Negative)	TN	FP
	Anomaly(Positive)	FN	TP

To compute the performance of ML algorithm we need to know the four terms [26]:

1. True Positives(TP): The positive instances are correctly classified by the learning algorithm

2. True Negatives(TN): The negative instances are correctly classified by the learning algorithm

3. False positives(FP): The negative instances are misclassified as positive by the learning algorithm

4. False Negatives(FN): The positive instances are misclassified as negative by the learning algorithm

From the confusion matrix, we can compute following metrics [8]:

1. True Positive Rate, TPR (also called Recall): Ratio of the anomaly instances correctly detected by the proposed method to all anomaly instances.

$$TPR = TP / (TP + FN) \quad (3.15)$$

2. False Positive Rate (FPR): Ratio of misclassified normal instances to all normal instances.

$$FPR = FP / (TN + FP) \quad (3.16)$$

3. Precision: Ratio of the anomaly instances correctly detected to all anomaly instances detected.

$$Precision = TP / (TP + FP) \tag{3.17}$$

4. Accuracy: Ratio of the instances correctly classified to all instances.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \tag{3.18}$$

5. F1 Score or F Measure: F1-scores is the harmonic mean of precision and recall.

$$F1\text{-}Score = 2*(precision*recall) / (precision + recall) \tag{3.19}$$

6. True Negative Rate (TNR): Ratio of actual negatives that are correctly identified as negatives.

$$TNR = TN / (TN+FP) \tag{3.20}$$

The objective of the anomaly detection is to increase TPR and decrease FPR, maintaining the good accuracy value.

3.7.2 Receiver Operating Characteristic(ROC) Curve

ROC curve represents the relation between the true positive rate and the false positive rate of a retrieval algorithm for each parameter value in one plot. The quality of a ROC curve is often summarized using the area under the curve (AUC). Higher the value of AUC scores better is the classification. A perfect classification has an area of 1.00 [25].

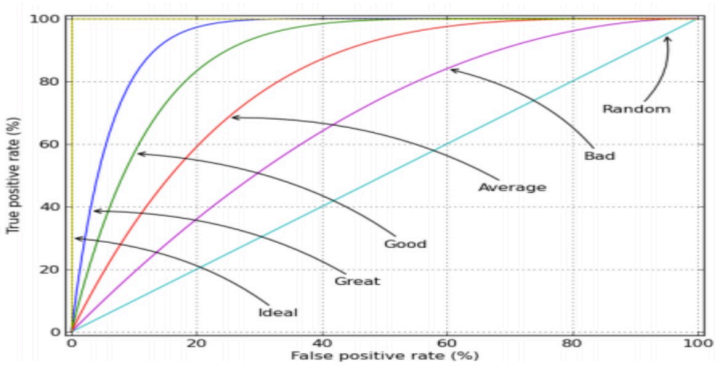


Figure 11. Illustration of ROC curve

3.8 Tools, Programming Language and Libraries

The following tools and programming language are used for this research work:

- Microsoft Excel
- Sublime Text
- Python Programming
- Tensorflow
- Keras
- Matplotlib

CHAPTER 4: EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Data Statistics

In the dataset CERTv4.2, over the course of almost 500 days, 1000 users generate 32,770,227 events (log lines). Among these raw events, 7320 of the events belong to anomaly. The data are processed from which 341794 feature vectors are constructed.

The first figure below shows the no of raw events analyzed vs no. of feature vectors constructed (sequences). The second figure shows no. of normal sequences vs no. of anomaly sequences.

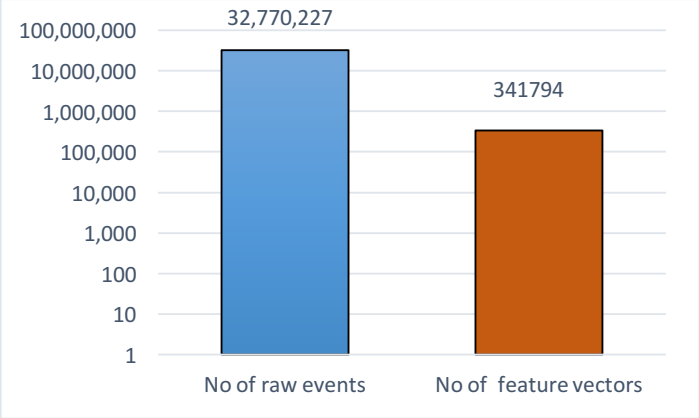


Figure 9. No of raw events analyzed vs no. of feature vectors constructed

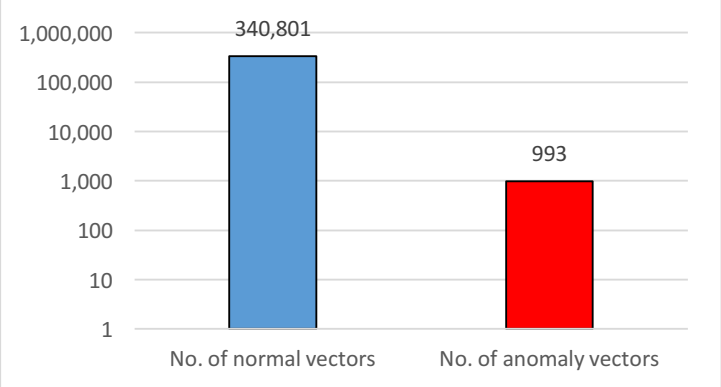


Figure 12. No of normal vs anomaly sequences

The following figure shows no. of anomalies belonging to each threat scenarios out of the total anomalies.

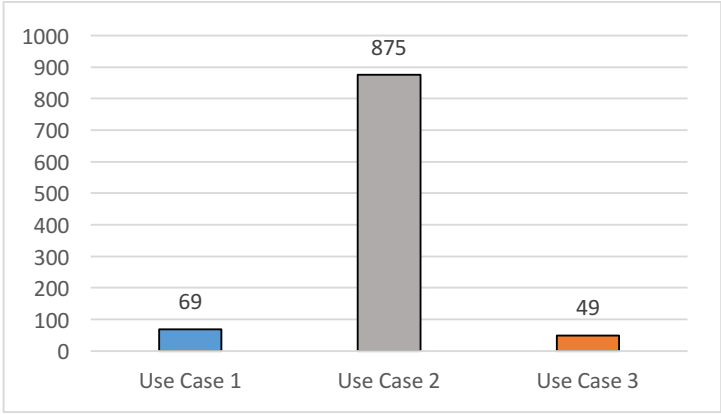


Figure 13. No. of instances in each threat scenarios

Each user's feature vectors are divided chronologically with time ordering into training, validation and testing in the ration of 70%, 10% and 20%.

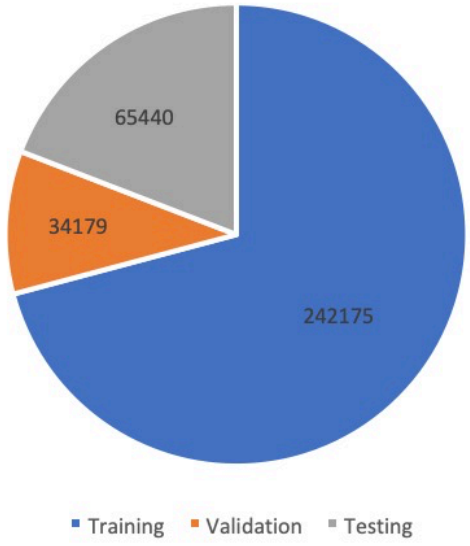


Figure 14. No of records in training, validation and testing

4.2 Experimental Results

4.2.1 Model Training and validation

The model was trained with training set (70% of total instances) and validated with validation set (10% of total instances). Following sections describe further details about the training phase.

1) Model summary

The input dimension and output dimension of autoencoder network is same. The total no. of features used is 44.

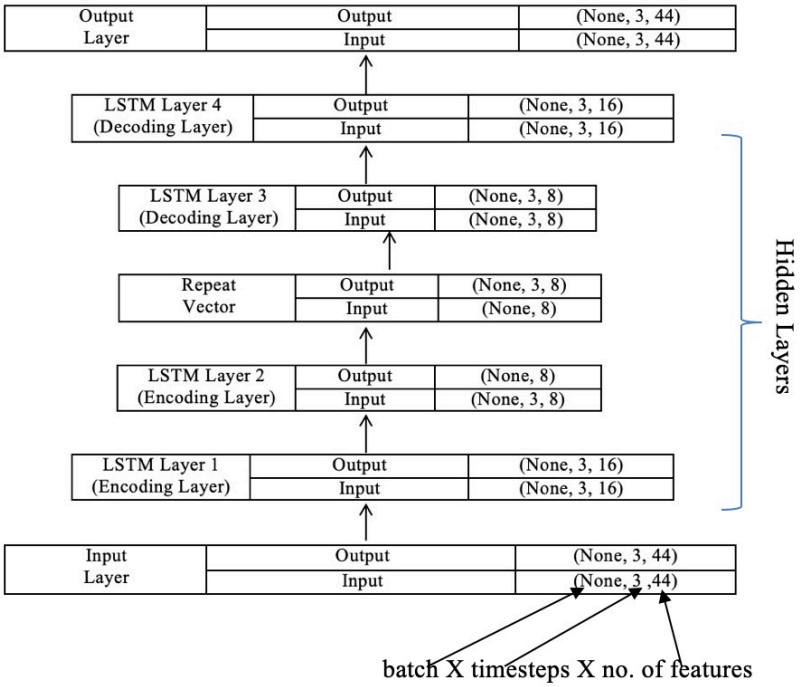


Figure 15. Model Summary

Note: In the above figure batch size is none, this is because batch size can be variable.
The model was trained with batch_size of 256, l2 regularization, learning rate: 0.0001, activation function: ReLU and Optimizer: Adam.

2) Model Loss

During the training phase, we pass the input sequence to the network and encoder encodes the information. Then decoder tries to reconstruct original information. To find difference between input sequence and output sequence, the "mean squared error" is calculated which is loss function. The loss is minimized during the training. In the basis of loss value the weights are updated in back-propagation. The loss should be minimized in the subsequent iterations. The following figure shows training and validation loss in each epoch.

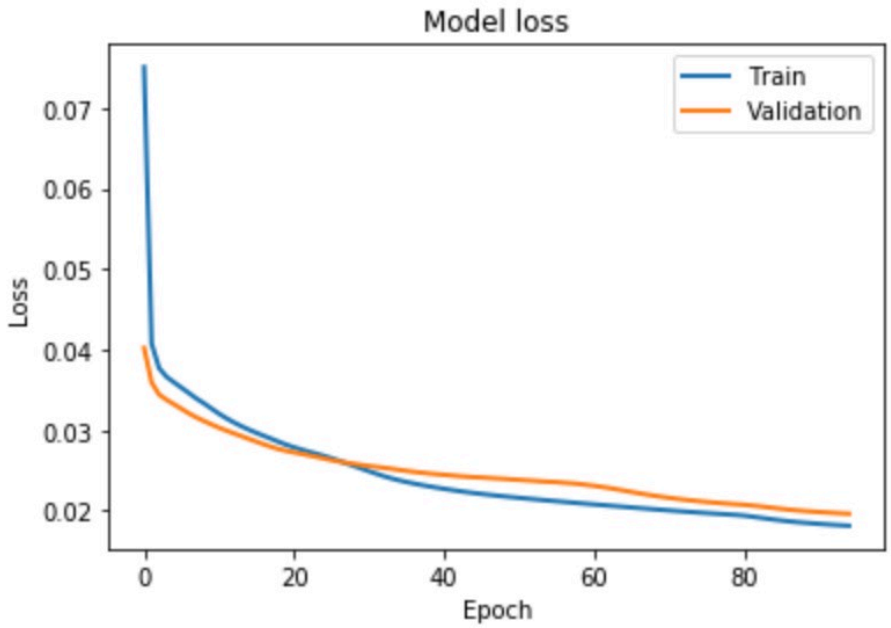


Figure 16. Train and validation loss during the training

3) Model training time

The model ran up to 95 epochs, and it was stopped when there is not significant difference in the validation loss value between successive iterations. Table below shows information about environment and training time

Table 6. Model Training Time

Environment	**Training Time**
macOS (cores 2, 8GB RAM)	Approx. 70 min

4.2.2 Model Evaluation

After model is trained we need to access effectiveness and performance of the model on test data. 20 % of the data are used for the testing purpose.

The model is trained in the normal data (or sequence), so the network learns normal behavior during training phase. The test data are actually labelled data; however, label is not sent to the model. The test data contain both normal as well as anomaly scenarios. Since model is trained in normal data, the model is expected to produce less error for normal data when we pass it through the model. For the anomaly data (which are unseen), the model is expected to produce high reconstruction error.

1) Choosing right threshold for classification

When reconstruction error is produced for each test data, the records are classified based on the reconstruction error. The sequences with reconstruction error greater than threshold value is classified as Anomaly sequence otherwise they are classified as normal. To choose right threshold value, TPR and TNR are plotted. The meeting point of these two curves is taken as a threshold value which produces high true positive and high true negatives.

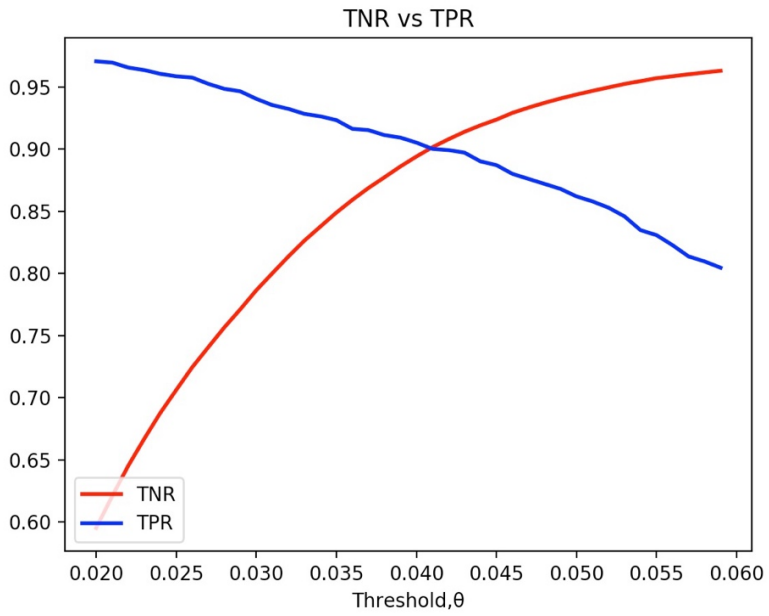


Figure 17. Choosing threshold value

2) Confusion Matrix

Table 7. Confusion Matrix

		Predicted	
		Normal	**Anomaly**
Actual	**Normal**	56035	6412
	Anomaly	94	899

From the confusion matrix, following performance parameters are calculated:

Table 8. Accuracy, TPR, FPR and TNR

Performance Metrics	**Percentage**
Accuracy	89.74
Recall (TPR)	90.53
False Positive Rate (FPR)	10.26
True Negative Rate (TNR)	89.73

The following chart shows graphical representation of Accuracy, TPR, FPR and TNR.

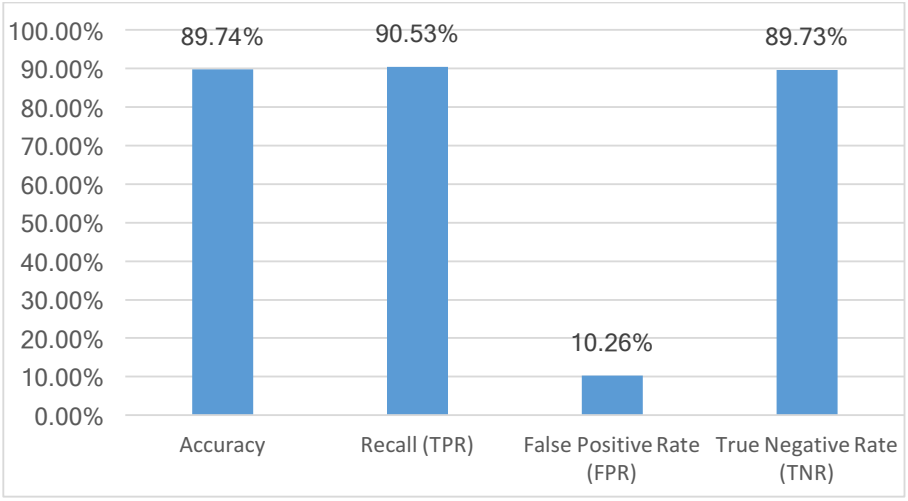


Figure 18. Accuracy, TPR, FPR and TNR

At different threshold values, the performance parameters can vary. Following curve shows variation in performance parameters at different threshold values.

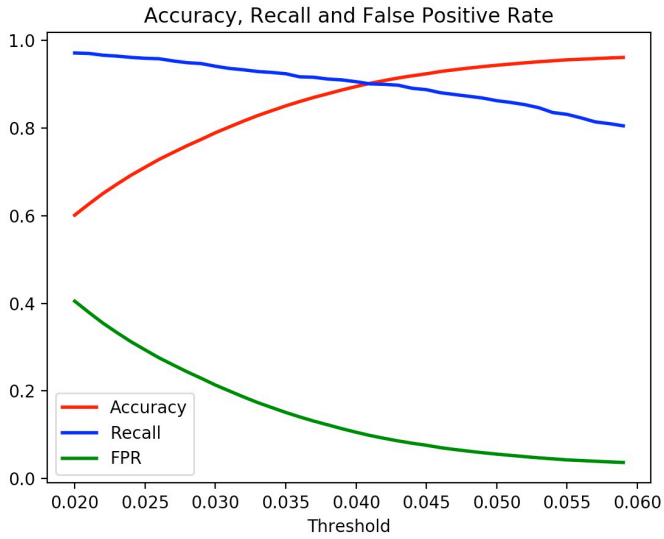


Figure 19. Accuracy, Recall and FPR at different threshold values

3) ROC Plot

The curve below represents the relation between the TPR and the FPR of the algorithm for each parameter value in one plot. The quality of a ROC curve is often summarized using the area under the curve (AUC). Higher the value of AUC scores better is the classification.

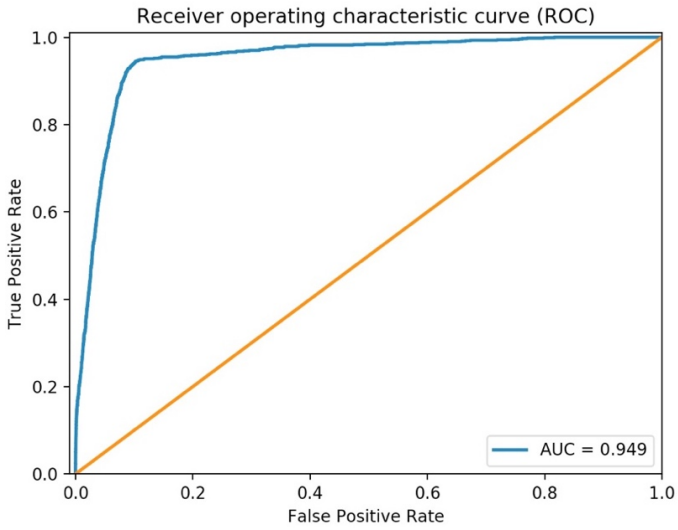


Figure 20. ROC Curve

4) Detection rate according to the anomaly scenarios

The following table shows the detection rate (percentage) according to the threat scenarios. Among the three scenarios the use case 1 has highest detection percentage.

Table 9. Detection rate according to use cases (threat scenarios)

Scenario	Total Instances	Correctly Classified	Detection Percentage
Use Case 1	69	67	97.1%
Use Case 2	875	788	90.06%
Use Case 3	49	44	89.8%

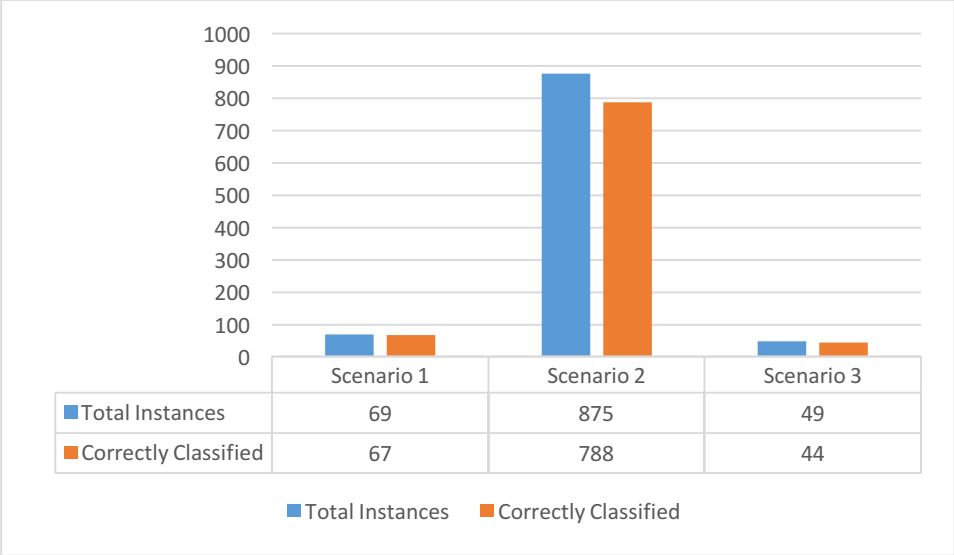


Figure 21. Detection rate according to threat scenarios

5) Experiment with different timesteps

No. of timesteps define how much past information to look back while producing current output. It defines how much information to hold in the memory. Following chart shows TRP and FPR at different timesteps.

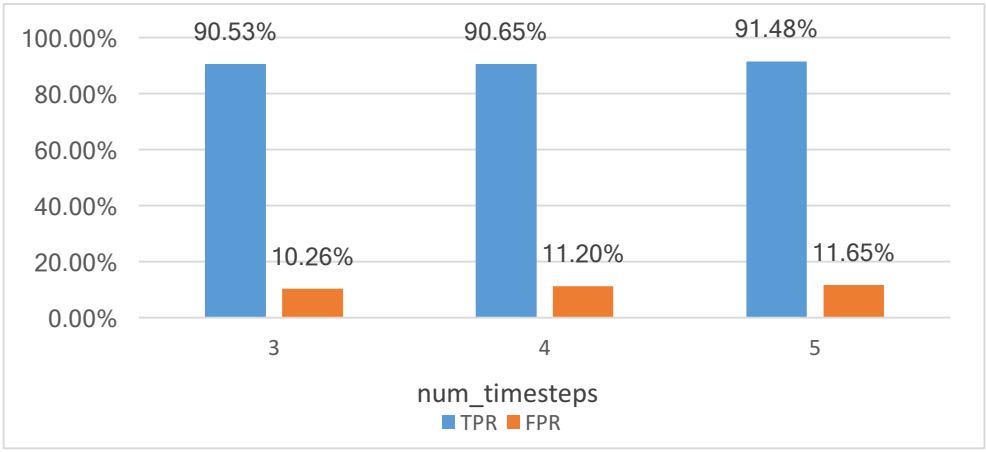
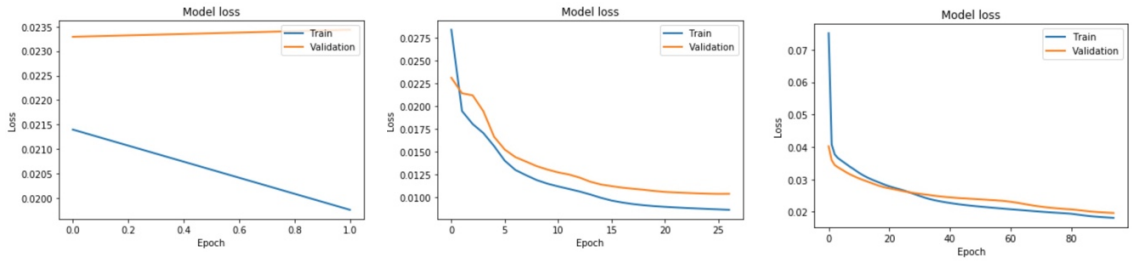


Figure 22. TPR and FPR with different timesteps

6) Experiment with Different learning rates

Learning rate define how fast weights change in the neural network. At varying learning rates, the Model Loss is plotted as shown below:



lr = 0.01 lr = 0.001 lr = 0.0001

Figure 23. Model loss at different learning rates

4.3 Analysis of the results

In this thesis work the LSTM Autoencoder; a semi-supervised deep learning approach has been implemented for user behavior profiling and anomaly detection. Looking at the CERT dataset, it is very unbalanced. Out of the total data, 97.7% are normal instances and only 0.03% are anomalous. The anomalies are extremely rare. So, the model is trained and validated only on normal instances. One challenge in the Autoencoder model is to find threshold for classifying the instances into anomaly and normal classes. Since, we want model to predict normal as normal and anomaly as anomaly, the meeting point of TNR and TPR is taken as the threshold value. The experimental results on test data shows that accuracy

of 89.74%, TPR 90.53% and FPR 10.26%. The main objective of anomaly detection is to increase TPR and decrease FPR. Since, anomaly data are very rare, we don't want to miss them. However, it was found that while increasing detection rate for positives cases, false positives are also increased.

There are three anomaly scenarios in the dataset. The detection percentage for Scenario 1(97.1%) is higher than Scenario 2 (90.06%) and Scenario (89.8%). This might be due to some field's value are more unique for Scenario 1.

Model performance was also evaluated by changing timesteps and learning rates. On increasing timesteps, true positives rate was found to be increased however false positives are also increased. On higher learning rates (0.01, 0.001) model stopped training more quickly and it didn't get converged. On too low learning rate (0.00001) model converge too slow. On moderate learning rate 0.0001, the results are found good.

After the classification of test data, some of the sequences are analyzed manually, to verify the results. In the following table, some of the cases for true positive, false positive and true negative have been listed.

Table 10. Result Analysis

SN	Sequence	Use Case	Actual class	Predicted Class	Explanation
1	{"MCF0600":{"logon_on_own_pc_normal":0,"logon_on_own_pc_off_hour":1,"logon_on_other_pc_off_hour":0,"logon_hour":23,"day_of_a_week":0,"device_connects_on_own_pc":0,"device_connects_on_other_pc":0,"device_connects_on_own_pc_off_hour":1,"device_connects_on_other_pc_off_hour":0,"documents_copy_own_pc":0,"exe_files_copy_own_pc":0,"exe_files_copy_other_pc":0,"documents_copy_other_pc_off_hour":0,"exe_files_copy_own_pc_off_hour":1,"neutral_sites":0,"job_search":0,"hacking_sites":0,"neutral_sites_off_hour":1,"hacking_sites_off_hour":1,"total_emails":0,"int_to_int_mails":0,"int_to_out_mails":0,"out_to_int_mails":0,"out_to_out_mails":0,"external_recipients":0,"mails_with_attachments":0,"role":"ProductionLineWorker",…..}}	1	Anomaly	Anomaly	The user logged after hour, he connects external device and he/she visits the hacking site.
2	{"CCL0068":{"logon_on_own_pc_normal":1,"logon_on_own_pc_off_hour":0,"logon_on_other_pc_off_hour":0,"logon_hour":8,"day_of_a_week":0,"device_connects_on_own_pc":2,"device_connects_on_other_pc":0,"device_connects_on_own_pc_off_hour":0,"device_connects_on_other_pc_off_hour":0,"exe_files_copy_own_pc":0,"exe_files_copy_other_pc":0,"exe_files_copy_own_pc_off_hour":0,"neutral_sites":62,"job_search":38,"hacking_sites":0,"neutral_sites_off_hour":0,"hacking_sites_off_hour":0,"total_emails":9,"int_to_int_mails":5,"int_to_out_mails":0,"out_to_int_mails":0,"out_to_out_mails":4,"internal_recipients":8,"external_recipients":6,"mails_with_attachments":2,…..}}	2	Anomaly	Anomaly	The user is visiting job hunting sites and using usb drives at higher rates.
3	{"BSS0369":{"logon_on_own_pc_normal":1,"logon_on_own_pc_off_hour":0,"logon_on_other_pc_off_hour":1,"logon_hour":7,"day_of_a_week":3,"device_connects_on_own_pc":7,"device_connects_on_other_pc":0,"device_connects_on_own_pc_off_hour":0,"device_connects_on_other_pc_off_hour":0,"documents_copy_own_pc":2,"exe_files_copy_own_pc":1,"exe_files_copy_other_pc":0,"documents_copy_other_pc_off_hour":0,"exe_files_copy_other_pc_off_hour":1,"neutral_sites":84,"job_search":0,"hacking_sites":1,"neutral_sites_off_hour":0,"hacking_sites_off_hour":0,"total_emails":12,"int_to_int_mails":10,"int_to_out_mails":2,"out_to_int_mails":0,"out_to_out_mails":1,"internal_recipients":14,"external_recipients":6,"distinc	3	Anomaly	Anomaly	The IT Admin has logged on to other pc after hour and he/she has connected usb and copied files.

4	{"CAH0936":{"logon_on_own_pc_normal":0,"logon_on_own_pc_off_hour":1,"logon_on_other_pc_off_hour":0,"logon_hour":4,"day_of_a_week":2,"device_connects_on_own_pc":0,"device_connects_on_other_pc":0,"device_connects_on_own_pc_off_hour":1,"device_connects_on_other_pc_off_hour":0,"documents_copy_own_pc":0,"exe_files_copy_own_pc":0,"exe_files_copy_other_pc":0,"documents_copy_other_pc_off_hour":0,"exe_files_copy_own_pc_off_hour":0,"neutral_sites":0,"job_search":0,"hacking_sites":0,"neutral_sites_off_hour":13,"hacking_sites_off_hour":1,"total_emails":0,"int_to_int_mails":0,"int_to_out_mails":0,"out_to_int_mails":0,"out_to_out_mails":0,"internal_recipients":0,"external_recipients":0,"distinct_bcc":0,"mails_with_attachments":0,"after_hour_mails":0,"role":"ComputerScientist"...	1	Anomaly	Normal	The user is logged on after hour and has connected the device. However, the model is not able to detect the anomaly.
5	{"HPH0075":{"logon_on_own_pc_normal":1,"logon_on_own_pc_off_hour":1,"logon_on_other_pc_off_hour":0,"logon_hour":7,"day_of_a_week":5,"device_connects_on_own_pc":6,"device_connects_on_other_pc":0,"device_connects_on_own_pc_off_hour":1,"device_connects_on_other_pc_off_hour":0,"documents_copy_own_pc":20,"exe_files_copy_own_pc":0,"exe_files_copy_other_pc":0,"documents_copy_other_pc_off_hour":0,"exe_files_copy_own_pc_off_hour":0,"neutral_sites":173,"job_search":0,"hacking_sites":0,"neutral_sites_off_hour":7,"hacking_sites_off_hour":0,"total_emails":19,"int_to_int_mails":8,"int_to_out_mails":1,"out_to_int_mails":0,"out_to_out_mails":11,"internal_recipients":13,"external_recipients":19,"distinct_bcc":2,"mails_with_attachments":8,"after_hour_mails":0,"role":"MechanicalEngineer"...	-	Normal	Anomaly	Normal sequence is misclassified as anomaly.

Note: Some of the feature are not shown in above table to save the table space.

4.4 Comparison with other research

Following table shows comparison of results with results on two papers done in the same dataset.

Table 11. Comparison Table

SN	Paper	Model	Accuracy	TPR	FPR
2	[10]	Distance Measurement	N/A	80	N/A
3	[23]	DBN-OCSVM	87.79	81.04	12.18
4	Proposed	LSTM Autoencoder	89.74	90.53	10.26

CHAPTER 5: CONCLUSIONS AND FUTURE ENHANCEMENTS

5.1 Conclusions

This thesis work presents a method to collect, process and analyze raw events for behavioral analytics and anomaly detection. It also suggests a method for key information extraction from the huge details of raw events, and prepare the feature vectors for ML. The CERTv4.2 dataset has been used for the research work. The dataset has very rare anomaly records and a huge number of normal records. Since anomalies are very rare, they are only used in the testing phase. The model was trained to learn normal behavior of users. For this, the unsupervised approach; LSTM Autoencoder has been implemented. The mean squared error between input and output is minimized during the training phase. During the testing phase, the model is expected to produce low MSE for normal behavior and high MSE for anomalous behavior. Each test data is fed to the network and calculated the reconstruction error. And the test data are labelled as normal or anomaly based on the reconstruction error. The predicted class and actual class of test data are recorded to the confusion matrix. The model performance is evaluated based on different metrics. The evaluation result in testing data shows that the model has True Positives 90.53%, False Positive 10.26% and Accuracy 89.74%. The objective of anomaly detection is to increase true positives and decrease false positive rate. However, it was found that while increasing true positives, false positives are also increased and accuracy decreased. The performance parameters; true positive rate, false positive rates and accuracy etc. can be controlled by choice of threshold value of classification. We can relate choice of threshold value with the analyst's time and budget to search for the anomaly. If the analyst has limited time or budget, he/she will set a threshold high and look for top activities where false positives are minimum.

5.2 Future Enhancements

While doing this thesis work, a lot of effort have also been spent in feature engineering. While extracting features from raw data, sometimes some key information can be missed. So, future enhancement could be applying deep learning algorithm in each log lines so that burden of feature engineering will be reduced and model itself learns the features that are necessary for profiling the behavioral pattern.

REFERENCES

[1] "Insider Threat Report," Crowd Research & Cybersecurity Insiders, 2018.

[2] Aaron Tuor, Samuel Kaplan and et.al, "Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams," in *Proceedings of AI for Cyber Security Workshop at AAAI 2017*, 2017.

[3] Gartner, [Online]. Available: https://www.gartner.com/doc/2831117/market-guide-user-behavior-analytics.

[4] Derek Lin, Baoming Tang and Qiaona/Joanna Hu, "Anomalous User Activity Detection in Enterprise Multi-Source Logs," in *Researchgate*, 2017.

[5] "The Essential guide to User Behavior Analytics," BALABIT Publication, 2016.

[6] "About Edward Snowden," [Online]. Available: https://en.wikipedia.org/wiki/Edward_Snowden .

[7] Xiangyu Xi, Tong Zhang, Guoliang Zhao and et.al, "Method and System for Detecting Anomalous User Behaviors: An Ensemble Approach," 2018.

[8] Fanzhi Meng, Fang Lou, Zhihong Tian and et.al, "Deep Learning based Attribute Classification Insider Threat Detection for Data Security," *2018 IEEE Third International Conference on Data Science in Cyberspace,* 2018.

[9] Madhu Shashanka, Min-Yi Shen, Jisheng Wang, "User and Entity Behavior Analytics for Enterprise Security," in *IEEE International Conference on Big Data (Big Data)*, 2016.

[10] Owen Lo, William J. Buchanan and et. al, "Distance Measurement Methods for Improved Insider Threat Detection," *Hindawi Security and Communication Networks,* vol. 2018, 2017.

[11] Tabish Rashid, Ioannis Agrafiotis and Jason R. C. Nurse, "A New Take on Detecting Insider Threats: Exploring the use of Hidden Markov Models".

[12] "Insider Threat Dataset," [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099.

[13] "THE CISO'S GUIDE to Machine Learning & User and Entity Behavioral Analytics," Aruba IntroSpect, 2017.

[14] B. Scho¨lkopf, R. C. Williamson and et.al, "Support vector method for novelty detection," *Advances in neural information processing systems,* p. 582–588, 2000.

[15] Mark Crovella,M. Ahmad Bashir and et.al, "Towards Detecting Anomalous User Behavior in Online Social Networks".

[16] "Deep Learning," [Online]. Available: http://deeplearning.net.

[17] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee, "Recent Advances in Recurrent Neural Networks," *arxiv,* 2018.

[18] "Understanding LSTM Networks," [Online]. Available: https://canvas.stanford.edu/files/1090785/download?download_frd=1.

[19] R. Can Aygun and A. Gokhan Yavuz, "Network Anomaly Detection with Stochastically Improved Autoencoder Based Models," in *International Conference on Cyber Security and Cloud Computing*, 2017.

[20] Xuan-Hien Le and Hung Viet Ho et. al., "Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting," *Water,* Vols. 11(7), 1387, no. doi:10.3390/w11071387, 2019.

[21] "CERT," [Online]. Available: https://wikis.ece.iastate.edu/insider-threat-detection/index.php/CERT.

[22] Fangfang Yuan et al., "Insider Threat Detection with Deep Neural Network," in *International Conference on Computational Science*, 2018.

[23] Lingli Lin et al., "Insider Threat Detection Based on Deep Belief Network Feature Representation," in *2017 International Conference on Green Informatics (ICGI)*, 2017.

[24] Iffat A. Gheyas and Ali E. Abdallah, "Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis," 2016.

[25] Mirza, A. H., & Cosan, S., "Computer Network Intrusion Detection Using Sequential LSTM Neural Networks Autoencoders," in *Signal Processing and Communications Applications Conference*, 2018.

[26] S. Sigdel, "An Approach to Develop the Hybrid Algorithm Based on Support Vector Machine and Naïve Bayes for Anomaly Detection," 2017.

[27] "Description of security events," [Online]. Available: https://support.microsoft.com/en-us/help/977519/description-of-security-events-in-windows-7-and-in-windows-server-2008.

[28] Zhaoli Liu, Tao Qin and et.al , "An Integrated Method for Anomaly Detection From Massive System Logs," *IEEEAccess,* Vols. VOLUME 6, 2018, 2018.

[29] "Nxlog," [Online]. Available: https://www.loggly.com/docs/logging-from-windows/.

[30] Joshua Glasser and Brian Lindauer, "Bridging the Gap: A Pragmatic Approach to Generating Insider Threat Data," *IEEE*, *Security and Privacy Workshops*, 2013.