INSIDER ATTACK DETECTION USING DEEP LEARNING

By

Rida Nasir

A thesis submitted to the faculty of Information Security Department,
Military College of Signals, National University of Sciences and Technology,
Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in
Information Security

October 2020

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by **Rida Nasir** Registration No. **00000203627**, of **Military College of Signals** has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have been also incorporated in the said thesis.

Signature: _____

Name of Supervisor: Dr. Mehreen Afzal

Date: _____

Signature (HOD):_____

Date: _____

Signature (Dean/Principal):_____

Date: _____

DECLARATION

I hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

This thesis is dedicated to

MY MOTHER, FATHER, AND BROTHER; SAAD

for their love and great endurance

ACKNOWLEDGEMENTS

All praises be for you ALLAH; al-Hadi, al-Fattah, and al-Muizz

As my supervisor, I'm grateful to Dr. Mehreen Afzal for many things. Notably, for creating the research environment in which I have performed my graduate studies. I requested her to supervise me, for following reasons: the areas of research in which she is involved,her professionalism and dedication towards her students and what I had heard about her good and friendly personality. I have not been disappointed in either regard. She has provided guidance at key moments in my work while also allowing me to work independently the majority of the time.

I'm thankful to my co-supervisor Dr. Seemab Latif and committee members;Asst Prof Dr. Hamad and Asst Prof Mian Muhammad Waseem Iqbal for their support and guidance.

I'm also obliged to HoD and all faculty members of Information Security department, who have influenced and enhanced my research

Finally, I would take this opportunity to show my deep gratitude to my parents, siblings, friends, and coursemates who see good in me and always push me to be the best version of myself.

RIDA NASIR

ABSTRACT

In today's world the most dangerous security threats are not launched by malicious outsiders or malware but from trusted insiders. The exploitation and leakage of sensitive data and information by malicious insiders is getting worse day by day. According to "Insider Report 2018" 90% of the organizations are prone to insider attack. Around 33% organizations encountered insider attacks in the last 12 months. However, most recent advancements and research in this field focus on using machine learning techniques for the detection of insider attacks because clues of malicious behavior of an employee may be extended over multiple datasets, concealed among hundreds of thousands of other data points, or mixed with normal user behavior, or separated by weeks or months of idleness. Now a days deep learning is a trending research topic and it is being applied in various security frameworks due to its enormous advantages.

Deep learning has enormous advantages. The algorithms outperform traditional machine learning algorithms in both performance and accuracy. However, there is no prior in-depth research in the field of insider attack detection. Insider attack detection using deep learning is still an open challenge. So the purpose of our research is the detection of insider threat by proposing a deep learning based novel approach **LSTM-AutoEncoder**. The proposed approach is compared to other techniques in terms of **Accuracy, Precison and F1 Score** and produces significant results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

DEFINITION	ACRONYM
Long Short Term Memory	LSTM
Recurrent Neural Network	RNN
Comma Seperated Values	CSV
Convolution Neural Network	CNN
Machine Learning	ML
Deep Learning	DL
Intrusion Detection System	IDS
Intrusion Prevention System	IPS
Area Under the Curve	AUC
Receiver Operator Characteristic Curve	ROC
Security Information and Event Management	SIEM
International Electrical and Electronics Engineering	IEEE
Carnegie Mellon University Computer Security Incident Response team	CMU-CERT
Support Vector Machine	SVM
Hidden Markov Model	HMM
Distance Measuring	DM
Principal Component Analysis	PCA
Guassian Mixture Model	GMM
Lightweght Directory Access Protocol	LDAP
True Positive	TP
False Positive	FP

Chapter 1

INTRODUCTION

One of the most basic, yet hard to solve question in cybersecurity is the identification of adversarial behavior. Due to recent technological advancements in the field of "World Wide Web" the geographical boundaries are not an obstacle anymore. But "every good thing comes at a price", due to this increased connectivity we are no longer able to distinguish the person we are interacting is a "friend" or "foe". Attacks on people, companies and leaderships can be implanted at any place on the web and rendered to cause damage of exceptional scale. So the knowledge of different types of attacks, consequences of their successful implementations are necessary. The most damaging attacks are the ones that are persistent and concealed for a long period [2]. These attacks then damage communal, economical and administrative infrastructure. On an ample scale, the general elections or referendum of a country can be improvised or fudged [10]. On a minute scale, the end-results of a company's delicate process can be influenced by the attacker [9]. A resistance against external or outside attacks are set up by most companies and governments but in today's world the most dangerous security threats are not launched by malicious outsiders or malware but from trusted insiders. The exploitation and leakage of sensitive data and information by malicious insiders is getting worse day by day. According to "Insider Report 2018" 90% of the organizations are prone to insider attack [1]. Around 60% organizations encountered one or more insider attacks in the last 12 months [6].

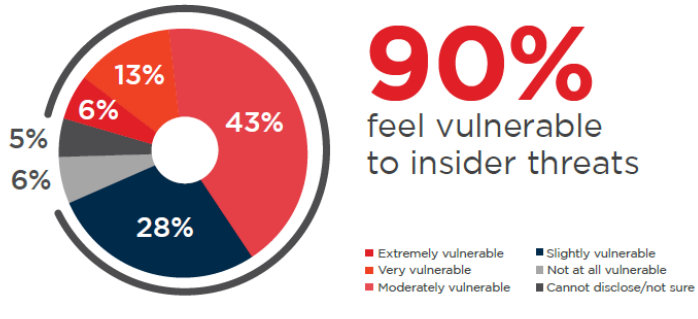

Figure 1.1: Insider attack statistics [1]

So what is an "Insider" and "Insider Threat", an insider is an authorized person which is allowed to access confidential resources within an organization, while an insider threat is a deliberate action by an insider that puts the organization at risk [2].
The insider threat can be categorized into three types [25, 26].

a. Disclosing company's critical assets to threats due to improper use of personal computers and remote devices.

b. Leakage or theft of sensitive data and information

c. The issue of cross-authorization, in which employees are given access rights to use any available machine, thus violating the access rights policies.

The crimes organized by Insiders requires multiple insiders from various departments of an organization. They remain hidden by evading the security mechanism deployed. Such users/persons are termed as active [12] insiders. An insider may impersonate and cause physical harm [11]. A passive insider is the one [12] who would only give data and statistics.

1.1 DETECTING INSIDER ATTACKS

The most recent advancements and research in the field of insider threat detection focus on using ML techniques for the detection of insider attacks because clues of malicious behavior of an employee may be extended over multiple datasets, concealed among hundreds of thousands of other data points, or mixed with normal user behavior, or separated by weeks or months of idleness. These datasets are extremely challenging and impractical for analyst to process and analyze in finding anomalous behavior.
Each user's behavioral data during a specific period of time (date, day, hour etc.) needs to be changed into numerical vector because machine learning algorithms takes continuous values as input. A user's behavior can be drawn out from multiple sources such as system logs, web URLs, email contents, network logs etc. Defining proficient features and changing this amorphous data into structured and organized dataset is the key factors in building a successful insider threat detection mechanism [22]. However, machine learning on the other hand outshines at detecting and predicting patterns from large datasets. Algorithms are developed to detect anomalies which are either the divergence from normal computing behavior or pol-

icy violations [3]. ML algorithms are mostly faster, more accurate and useful in protecting against dangerous risks.

The learning model-based approach is favorable because it does not have any dependency on the domain professional knowledge for establishing rules or creating relational graphs but it has two considerable and practical limitations:

1. The way of estimating a user's behavioral data.
2. Insufficiency of anomalous cases required for model building.

1.2 RESEARCH QUESTION

In this this thesis, we want to analyze how deep learning can be applied to user's technical data inside an organization. Moreover, we want the system to be simple, adaptable and minimum domain knowledge requirement.

Based on the above discussion below mentioned research questions are devised.

1. How deep learning can be utilized for the efficient detection of insider threats with minimum domain knowledge required?
2. How can the devised technique produce efficient results with high accuracy and low false positive rate and be applied with minimum resources?

1.3 RESEARCH OBJECTIVES

The objectives to carry out this research are as follows:

1. To study and analyze previous and on-going insider attack detection approaches using deep learning.
2. To propose a deep learning based efficient insider attack detection scheme that will detect insider attacks with more accuracy and low false positives.
3. Evaluation and comparison of the proposed scheme with previous approaches.

1.4 PROPOSED SCHEME

Proposed scheme to carry out this research is as follows:

1. Data gathering from various sources.

2. Data cleaning to have consistency.

3. Model building (choosing right DL algorithm).

4. Gathering insights from model results.

5. Data visualization, transforming results into visual graphs.

6. Comparison of results.

1.5 RESEARCH METHODOLOGY

The methodology adopted in this thesis consists of five stages [27]:
Understanding, Proposition, Implementation, Assessment and Conclusion.
During understanding phase, relevant literature is reviewed. During the proposition phase, the scheme to contribute to the current situation is proposed.The proposed scheme consists of several steps and is described in detail in 1.4. In the course of implementation phase, the proposed scheme is implemented. Later, the developed model is evaluated with performance measures. Finally, the results are presented with a conclusion.

1.6 ORGANIZATION

The remainder of this document is organized as follows: Chapter 2 reviews the recent literature related to insider threat detection. Chapter 3 is composed of our data selection procedure, data preprocessing step and proposed detection algorithm for implementation. Chapter 4 represents the experimental evaluations of our proposed detection scheme. Chapter 5 concludes the document.

Chapter 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter explains in detail all the terms, definitions and literature relevent to Insider Threat Detection using ML and DL techniques.

2.2 TYPES OF INSIDERS

2.2.1 Inadvertent Insiders

They are the careless employees that do not pay much attention to the trainings and company's IT policies and mostly causes fault by mistake. They are also known as negligent insiders. These employees inadvertently due to their laziness and lack of conscientiousness cause security violations by accidental disclosure, mishandling of systems, or downloading malware [7]. According to [6] 70% of the data breaches are inadvertent data breaches.

2.2.2 Malicious Insiders

They are the ones that takes edge of their inside position in order to attain personalize benefit through illicit means. They either gained the job for malevolent motives or has been offered some financial incentive for committing insider attacks on account of a rival or a competitor. These insiders can be a part of a criminal faction or may be involved in the sale of business secrets. The employees that are planning to leave their job or starting their own setup also deliberately steals data for their personal gain and also comes under the category of malicious insiders [8] .62% of the data breaches are caused by malicious insiders [6].

2.2.3 Disgruntled Employees

They are the ones who commits vandalism and data breaches intentionally [7]. Unappreciated or being overlooked by the authorities serves as the main motivation for disgruntled insiders. Employees who feel betrayal of some sort by the organization, management, or a colleague might develop the feelings of revenge resulting in an attempt to destroy organization's stature by planting a "logic bomb" inside the organization before quitting the job [8].

2.2.4 Vendors and Contractors

They sometimes act as insiders because most often they have privileges to sensitive systems and confidential data that can be traded or jeopardized. These workers are not the official employees of the company, due to which they might not be loyal to the organization causing them to behave carelessly about data security practices [8]. 52% of the insider threats are caused by contractors, service providers or temporary workers [6].

2.3 TECHNIQUES USED BY INSIDERS

The main edge insider attackers have over external attackers is the ability to bypass security checks and remain undiscovered and poses serious damage to the organizational assets. Insiders know where valuable and critical assets are, and what is valuable. They are given authorized access to these assets and the ways in which the confidentiality, availability, or integrity of data can be undermined. The techniques used by an insider to cause security breach are as follows [2]:

- Obtaining of user credentials by social engineering either by searching the database for information or convincing a pervious employee for information. The obtained information can later be used for insider trading.
- By using a legitimate system for stealing records.
- Bypassing of security procedures
- Using of end point IT devices to launch insider attacks [13].

2.4 MOTIVATIONS FOR INSIDER ATTACKS

Various primary and secondary elements that serves as an insider attack inspiration includes financial gain or greed, revenge, anger, thrill, pressure, treachery, discontentment, jealousy, organizational politics and acknowledgement [15]. Some general motivations for an insider to damage an organization are listed in [13] which includes:

1. Fraud (57%) [6]
2. Financial gain (50%) [6]
3. Sabotage (41%) [6]]

4. Espionage (32%) [6, 14]

5. IP theft (43%) [6]

6. Professional gain (11%) [6]

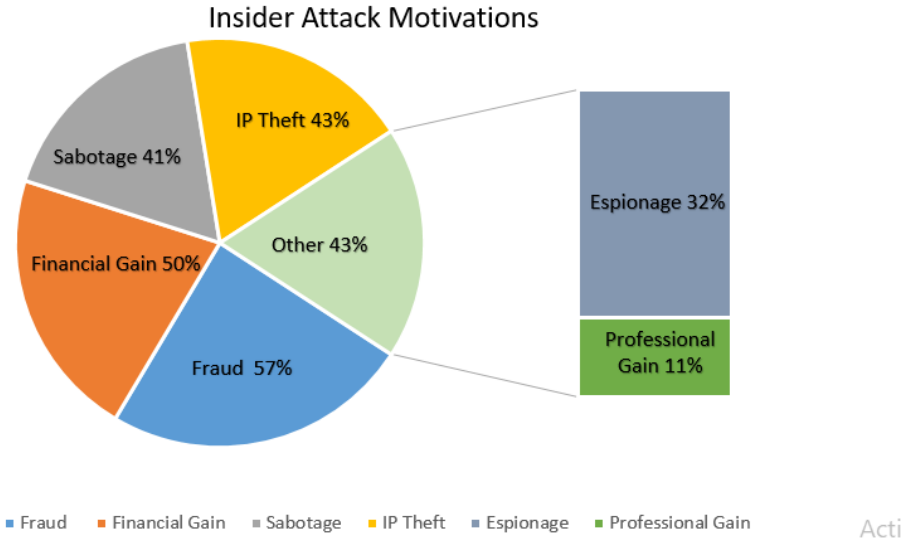

Figure 2.1: Insider Attack Motivations

2.5 THE INSIDER KILL CHAIN

The concept of insider kill chain is first used by [29] . We will be able to protect our potential assets against successful attacks by developing strategies at each stage of the kill chain. [30, 31] .

The insider kill chain contains five stages:

- **Recruitment**

  The initial step is where a reliable insider becomes malevolent. The potential factors may include financial gain, greed, wanting to start their own setup etc. a trusted insider started concealing connections and communications with outside third parties could be an alert [28].

- **Reconnaissance/ Information gathering** The malicious insider will search for important and valuable information. The time spent at this stage depends on the knowledge and skill set of the insider [28].

- **Manipulation**

  In this stage, the insider exploits the access rights given to him, to gain access to critical assets i-e. file server, database server etc [28].

- **Obfuscation**

  The insider will then try to bury its trails, which includes actions like modifying file names and deleting web history and cookies [28].

- **Exfiltration**

  Transferring the information outside the organization serves as the last step of insider kill chain. The process can be done either by using CD, USB, email, via network or file sharing [28].

Figure 2.2: Insider Kill Chain

2.6 ANALYTICAL FACTS

Mostly "Insider Threat" is associated with malicious employees which aimed to harm the company by theft and vandalism, however in actual, negligent and careless employees often accidently causes a high impact damage (66%) [6]. The biggest insider threat actors are regular employees (49%) and privileged IT users (59%) [6] followed by contractors (52%) [6] . Around 57% of the confidential business information, 52% account information and 49% private data is lost during 2018 as a result of insider attack [1]. However in 2019 customer data is considered to be most vulnerable (63%) [6]. IT assets which are most prone to insider attacks include databases (50%) and file servers (46%) [1].

The main enablers of insider attack are the increased number of users with unnecessary access rights, increasing devices accessing confidential data, technological complexities increasing rapidly, lack of user awareness and training and increase in sensitive data [1]. The

most upsetting truth is that the potential loss caused by successful insider attack ranges from $100,000 to $500,000. Due to such a high impact, around 31% organizations consider insider attacks more damaging and destructive in contrast with external attacks [1]. 85% of the organizations finds it difficult to calculate the real harm caused by a successful insider attack [6].

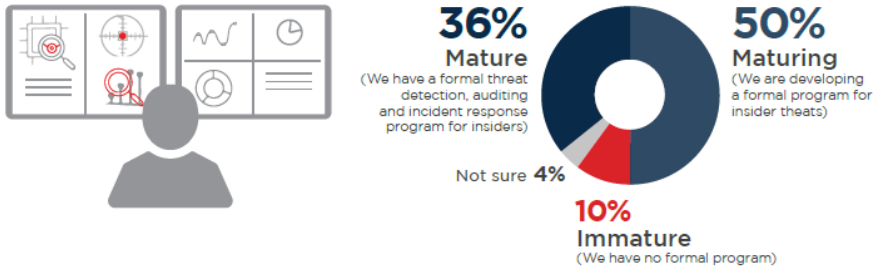

Figure 2.3: Insider attack controls inside organizations [1]

2.7 DETECTING INSIDER ATTACK

For insider threat detection there exists three conventional approaches.

- **Rule-based Detection** Establishment of a rule based detection mechanism is the first approach [17, 18]. In this approach a rule set is created by a group of professionals to recognize malicious actions. Then, each user's conduct is logged and analyzed to identify whether it matches to any of the pre-written rules. Rule-based detection approaches have a picky restriction which is the constant update of rules through the knowledge of domain specialists, so the probability of someone bypassing the rules always exists [20]. Thus making rule-based detection approaches unadaptable to changing insider threat patterns [17, 20, 21].

- **Graph based Detection** Modeling a network graph to locate doubtful users or malevolent patterns by observing the remodeling of the graph structure is the subsequent approach for insider threat detection [22]. This approach analyzes both the data and its relationships. The relationships between the data are shown by edges tethering the graph nodes, by examining the properties relationships between specific nodes to insider threats can be identified. [23].

- **Machine Learning based model Detection** Building a numerical or machine learning based model to forecast prospective malevolent behavior based on prior information [24]. The main aim is the development of a mechanism that automatically detects users who performs peculiar activities amidst of all users without preliminary knowledge or rules. The detection performed by ML is more accurate as compared to rule-based approaches, because the algorithm learns, updates and adapt to the changing data [22].

2.8 MACHINE LEARNING

Machine learning is the branch of Artificial Intelligence (AI) which gives machine the ability to learn automatically without human interference and improve their learning with experience. Arthur Samuel is credited for inventing machine learning in 1959 [66]. Since 1959 ML has grown immensely and is used today in various end user and professional products. Biometric verification such as fingerprint and facial identification, voice recognition, recommender systems , e-mail spam filtering, search engines,disease detection are some of the uses of ML .It consists of two main phases, training and testing. During training, at first features and classes are defined inside training dataset, the features are reduced and used for classification, a model is created for learning and applied on training data. The model is trained on training data and then tested on test dataset in order to measure success rate [4].

Machine learning algorithms are mainly classified:

- Supervised Learning

- Unsupervised Learning

- Reinforcement Learning

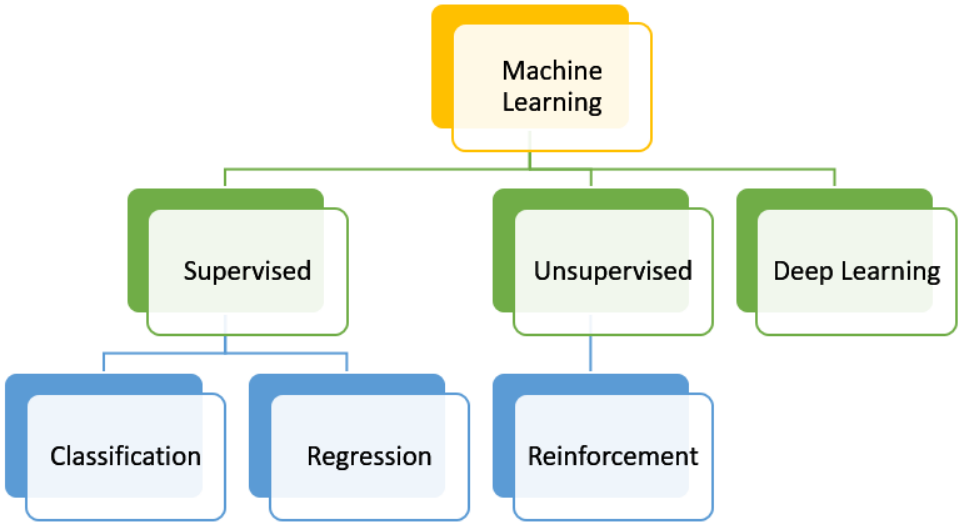

Figure 2.4: Machine Learning Classification

2.8.1 Supervised Learning

In supervised learning labelled data is used to train the algorithm. Labeled data indicate that it is already tagged with the correct answer. A good training dataset is analyzed by the algorithm and a function is modeled to make efficient future predictions.

Some of the common supervised algorithms include:

Decision Trees, K Nearest Neighbors, Linear SVC (Support vector Classifier), Logistic Regression, Naive Bayes and Linear Regression.

However, supervised learning can be less useful in scenarios when the input labeling is not possible. It does not take place in real time. Computational complexity and time intensiveness are also real challenges. Data labeling is very challenging when data size is dynamic and increasing continually. But the advantages and disadvantages of this type of learning mainly depends on the user's choice of algorithms. Each algorithm has its applications for example text classification, anomaly detection, natural language processing, image recognition etc. [4]

2.8.2 Unsupervised Learning

Unsupervised learning does not use predefined labels. The algorithm does not require human interference, the data is sorted on the basis of some resemblance and differences without any prior training [4]. Some of the common unsupervised data algorithms includes:

K-means clustering, K-NN (k nearest neighbors), Dimensionality Reduction, Principal Com-

ponent Analysis, Independent Component Analysis and Mixture models.
However, despite its various advantages, it is less accurate, results are difficult to confirm because of absence of labelled data [4]. No of clusters need to be specified in advance which is not an easy task. Insider attacks may be detected using unsupervised learning techniques, but the results produced can be less accurate and may result in higher false positive rate.

2.8.3 Reinforcement Learning

Reinforcement Learning is Similar to unsupervised learning. The algorithm is not provided with a solution. It is comparable to trial-and-error approach [68].

2.8.4 Deep Learning

Now a days deep learning is a trending research topic and it is being applied in various security frameworks due to its enormous advantages. it can be used in both supervised and unsupervised manner. DL is a subfield of machine learning. It uses a layered framework of algorithms called artificial neural networks, encouraged by the structure and function of human brain. Thus making these algorithms more intelligent as compared to other machine learning algorithms. One of the major advantage of DL is its ability to handle huge amount of data (big data). The performance improves as the data size increases. Also the performance of the algorithm improves with training, resulting in continuous increase in performance. The algorithms have the ability to achieve high accuracy. DL enables machines to solve complex problems even if the dataset is diverse, unstructured and inter-connected. The term "deep" refers to the number of hidden layers in between input and output layer of the neural network architecture. In deep learning features are extracted automatically and it provides "end to end" learning [5].

Some of the most common deep neural networks are CNN (convolutional neural network) RNN (recurrent neural networks) and MLP (multi-layer perceptron). Deep learning has enormous advantages. The algorithms outperform traditional machine learning algorithms in both performance and accuracy. It can be used to improve insider detection results with high accuracy and lower false positives, thus enabling organizations to have a robust insider threat detection mechanism.

2.8.5 LSTM

Long Short Term Memory commonly known as LSTM is a special type of Recurrent Neural Network (RNN), able of learning long-term dependencies. Introduced by Hochreiter Schmidhuber in 1997. This family and LSTM, especially, have the capability of automatically extarcting past events effects from sequential data LSTM has the potential of extracting both long and short- term concequences from previous events. It has a default behavior of remembering information for long intervals of time [83].

LSTM is suitable for classification, processing and predicting time series problems of unknown duration. Relative inconsideration to gap length makes LSTM advantageous over other RNNs varients, HMM and other sequence learning techniques. LSTM have a chain structure, but unlike it the repeating module has four layered distinct structure, interacting in a very special way.

The output of the current training stage can be saved and forwarded to the next input layer through an iterative structure, allowing the system to be influenced by the new record's behavioral patterns on top of historical records.The LSTM is one of RNN neural networks most powerful variants [78, 81]. Instead of tracking the effects of recent records that can be obtained by RNN, the LSTM recalls long-term dependence. In addition, LSTM has now been commonly used in machine translation [30], identification of anomalies [80] and self-diagnosis of medicine [79]

LSTM Architecture

LSTM architecture consists of a cell and three "regulators", known as gates. An input, output and a forget gate for information flow inside the LSTM unit: [84]. When a new value passes into the cell, it is controlled by input gate, the extent to which a value stays within the cell is controlled by forget gate. The amount of the cell value used to measure the LSTM unit's output activation is controlled by forget gate. The operational sigmoid function is used as gates activation finction [84].

2.8.6 Autoeocoders

Autoencoder is an unsupervised ANN algorithm that learns how to compress and encode data coherently and how to reconstruct the data from a reduced encoded characterization to a representation as close as possible to the actual input. By design, autoencoder reduces the

dimensions of data by learning how to exclude the data noise [81]. The ideal autoencoder model has the following two characteristics [82]:

- Capable to build a reconstruction accurately from inputs.
- Insensitive enough to the inputs that the model doesn't just learn or overfit training data.

Autoencoder Components

Autoencoders consists of four main components [81]:

1. **Encoder:** Here the model learns how input dimensions can be reduced and input data be compressed into encoded representation.
2. **Bottleneck:** Compressed representation of the data is incorporated here. This is the input data's lowest possible dimension.
3. **Decoder:** At this layer the algorithm learns how to regenerate the encoded data to be as similar to the genuine input.
4. **Reconstruction Loss:** This method evaluates the performance of deceoder and how similar the output is to the actual input. Back propagation is used in training to minimize the network's reconstruction loss.

Autoencoders Architecture

Figure 2.5: Autoencoder Architecture

Deep Autoencoder

Deep Autoencoders is comprised of two similar deep belief networks, one for encoding and another for decoding. Four to five layers are generally used for encoding and decoding. The model is pre-trained in an unsupervised layer by layer technique. The building block of deep-belief network is Restricted Boltzmann Machine. Various applications includes topic modeling, or statistical modeling [86].

Advantages [87]

- Deep autoencoders can be used on real-valued dataset.
- The final encoding layer is concise and quick.

Drawbacks [87]

- Due to increased number of parameters than input data there are chances of overfitting.
- The learning rate is made slower at the stage of the decoder's backpropagation, due to the type of data handled. Thus makes data training a problem.

2.8.7 LSTM Autoencoders

An LSTM Autoencoder is an implementation of an autoencoder for sequence data using an Encoder-Decoder LSTM architecture. An encoder-decoder LSTM is programmed to read the input sequence, encode it, decode it, and reproduce it. The ability to reproduce an input sequence is considered as a performance evaluation criteria for a model. LSTM auto-encoders are specifically designed to overcome the problem of long-term dependency, remembering information for long intervals of time is practically their default behaviour and hence they have an advantage over normal auto-encoders. The decoder part of the model may be removed, once the desired level of performance for recreating the sequence is achieved. Input sequences can then be encoded to a fixed-length vector by this model [88]. The steps followed are:

- an autoencoder is build on the normal (negatively labeled) data.
- It is used for reconstruction of a new sample.
- If case there is a high reconstruction error, it will be labeled as insider attack.

To deal with the problem of insider threat, multiple models and solutions are proposed by researchers. They proposed the abstract idea of insider threat and abstract model of insider behavior [32].

2.9 LITERATURE RELATED TO INSIDER ATTACK DETECTION USING ML AND DL TECHNIQUES

2.9.1 USER BEHAVIOR BASED INSIDER DETECTION TECHNIQUES

In [33] the researcher proposed a supervised learning technique for insider attack detection in which user activities are classified based on time series mechanism. At first user activities

for a single day are computed from user logs and then a time-series feature vector is established from the statistics of single-day features for a time period. And feature vectors are labeled as malicious and non-malicious. The imbalanced CMU-CERT dataset is adjusted using an adjustment algorithm. A two-layer deep autoencoder neural network is deployed for classification, and its performance is compared to random forest and multilayer perceptron, the proposed approach produces results with higher precision, recall and f-score. This paper is an improvement to some-existing techniques, and combines the advantages of both supervised and unsupervised learning. It uses time-series based user activity classification to deal with the changing user behavior over a time interval.

In [34] the researcher proposed a deep learning based anomaly detection technique, which is used for insider threat detection. LSTM-CNN algorithms are used to identify user anomalous behavior. At first LSTM is used to learn user behavior by monitoring user activities and extract temporal features, these features are then changed to fixed-size feature matrix and then fed to CNN for insider threat detection. The proposed approach produces good results with an AUC= 0.9449. However, experimental setup is not explained, the platform they used for evaluation of results, the memory and processing requirements, the time of training and testing the model are all missing. Only the dataset on which the experiment is performed is publically available CMU-CERT dataset V4.2.

In [35] the researcher proposed a user behavior based anomaly detection method. At first User behavior characteristics features are extracted from audit logs, then these features are used to train the detection algorithm XGBoost in order to detect insider threat. In order to deal with unbalanced data, a smote algorithm is used for dataset balancing. k-fold cross validation is used in order to avoid the problem of over and under fitting. The dataset on which the experiment is performed is publically available CMU-CERT dataset V6.2. The paper claims to be the first one to use XGBoost algorithm for insider threat detection. And achieves 100% recall, and produces good results as compared to random forest, multilayer perceptron and SVM.

In [36] the researcher proposed an insider attack detection scheme, in which user behavior is extracted from network logs. The network log files contain multiple user behavior over a period of time. Features and fields are extracted from these behavior logs for behavior auditing, and then these log files are used to train the Improved Hidden Markov Model

(IHMM) for detection of malicious behavior.

In [37] an insider threat detection scheme based on behavior analysis of individual user by analyzing its activities over a period of time is proposed. In order to reduce false positive, any deviation from expected user behavior is compared to its behavioral pattern and also with the behavioral model of whole system users. Various features are extracted and are ranked and prioritize based on their importance, for example policy decision is ranked as important because it gives the final decision. The feature extracted are fed to the Random Forest algorithm for detection and produces significant results and a low false positive rate. The evaluation of the proposed model is performed on both public (CMU-CERT) and private datasets (NextLabs). The algorithm used claims to produce 97.81% accuracy and only 2.19% error. The error rate of Random forest algorithm generally decreases as the model learns the data. The algorithm used is one of the most accurate and provides good results even with small number of training samples, there is no need for feature normalization and individual trees can be trained in parallel.

In [38] the researcher proposed a user sentiment analysis based insider attack prediction scheme, in which user's network browsing content and emails are used to build the sentiment profile. The proposed scheme predicts user attack motivation based on their daily and weekly threat profile, and generate an alert if the threat value is higher than a threshold value. Sentiment analysis module and malicious URL detection module is used for building user sentiment profile by using their network browsing content and email content. A dictionary based sentiment analysis method is used. Convolution neural network is used for detecting malicious URL. Threat score is computed for each user based on its daily activities I-e. the no of malicious URL websites visited, negative emails sent etc. For threat detection, mean and variance of the threat value is calculated. The anomaly score is calculated using user threat score, mean and variance. The dataset on which the experiment is performed is publically available CMU-CERT dataset V4.2. The dataset is extended by adding negative samples. For websites with negative information, Dark Net Archives is used. For malicious URL sharing websites, Virus Total is used. For email negative samples, twitter emotion corpus is used. Sentiment classification accuracy of 100% for http and 96% for email content is achieved. This model claims accurate prediction of malicious insider based on their attack motivation predictor.

In [39] the researcher proposed a deep learning based insider attack detection mechanism, in which LSTM algorithm is used to model user log as natural language sequence to achieve role based classification. LSTM is very effective in natural language processing. It can automatically learn features. It is very good in processing sequence data due to which it is used to model user behavior. The dataset used is CMU-CERT public dataset.

In [48] a framework known as "Insider Catcher" is proposed. The proposed framework uses LSTM, a deep learning technique to model system logs as organized sequence. User normal and adversarial behavior is detected by user behavior pattern analysis. The proposed framework claims to be more efficient than other log based systems and works on real time systems. The dataset used is CMU-CERT public dataset. Version of the dataset is not given.

In [41] the researcher proposed an insider attack detection scheme using distance measurement techniques (DL distance, Jaccard Distance and Cosine Distance) through analysis of user activity. The model is evaluated on CERT dataset and 3 insider attack scenarios were extracted and results are compared with HMM model. Jaccard method works well on scenario 2 while HMM model out performs the all three DM techniques. But DM techniques are faster than HMM. The evaluation is performed on synthetic data set which have uniform user activities.

In [43] the researcher proposed an insider attack detection scheme using Kernel PCA and LSTM-RNN. At first event aggregator is used to normalizes information from various events (http, email, system call etc.) into similar data types. Then feature vectors are extracted and used for training the model. Multiple classifiers fusion strategy is used to reduce the classification accuracy errors of classifiers. After attribute classification, the results are fed to the anomaly calculator which uses fusion decision algorithm to detect anomalous behavior. Kernel PCA is used to reduce high dimensionality problem. The dataset on which the experiment is performed is publically available CMU-CERT dataset V6.2. over tensor flow framework. The proposed scheme produces good results with a precision=95.12% and accuracy=93.85% as compare to isolation forest, SVM and PCA.

In [46] a deep autoencoder based anomaly detection methodology is proposed. Frequency based theory is used for obtaining feature. Particular type of audit data is used to train each autoencoder respectively and its perfect model is tuned. The reconstruction error between

the actual and decoded stipulate a difference between regular and abnormal user behavior. Final anomaly score is computed by combining the results of four separate detectors. The dataset used was CMU CERT V6.2. The researcher claims that the proposed scheme can detect all insiders with a relatively low number of false positives.

In [47] the researcher proposed a deep belief network (DBN) based composite methodology for insider threat detection. An unsupervised DBN for unseen feature selection, from multi-domain features obtained from logs is used as the first step. Later on, the extracted features from DBN are used to train a One-Class SVM (OCSVM. Feature engineering and processing using DBN is a new idea proposed by the researcher. The dataset used was CMU CERT V4.2. The performance of the technique is compared with an existing technique Pearson-OCSVM and evaluated in terms of accuracy which is 87.79% and false positive rate.The results produced shows the effectiveness of the proposed technique in insider threat detection.

In [50] the researcher proposed a new technique to cope with the problem of insider threat inside an organization. The proposed technique illustrates user behavioral data by examining shell commands flow, keystrokes, and mouse functions while interacting with GUI. User behavior features for sketching behavioral profiles are extracted using a probability distribution function. The performance is evaluated based on publically available dataset in terms of accuracy and false positives.

In [53] a new unsupervised detection technique that uses system logs to detect user's abnormal behavior is proposed. The technique uses denoising autoencoders for encoding user log file, and uses integrated methods to identify anomalous data. The integrated methods used include: GMM, buck covariance, OCSVM, isolation forest and local outlier factor. The anomalous behavior of the user is detected efficiently as compared to the conventional techniques. User characteristics and domain understanding is not required, and the technique is purely data driven.

In [55], two unsupervised anomaly detection algorithms are used and a contrast is drawn between different system logs comprising of both daily and frequently aggregated one. A concise feature set is extracted from system logs. A trust score from previously generated anomaly score of each user is fed to the next interval's model and show its effectiveness and

impression in detecting insiders. Moreover, the effectiveness of user's psychometric score is used in insider attacks prediction, and is claimed to be the first one to use user's psychometric score. The evaluation is performed on CMU CERT dataset V4.2.

In [56] user behavior profiling and anomaly detection based insider-threat detection method is proposed. From user logs, three types of datasets are formulated, user day-to-day activity overview, user's weekly email logs and email topic grouping based on content. Malicious behavior is then detected using four anomaly detection algorithms and their integration. The experimental results indicated that the proposed approach works well on unbalanced datasets and requires no prior domain knowledge.

In [57], a data based approach for insider threat detection is proposed. The extracted features include email content, work-patterns and web history. These features are then fed to two different approaches (i) an unsupervised anomaly detection model (ii) a supervised classification model. The proposed approaches prove significant in terms of detecting insider threats.

In [58] Multi State Long Short Term Memory (MSLSTM) and Convolution Neural Networks (CNN) based hybrid machine learning approach is proposed. The technique works by using time series anomaly detection method for outlier detection in user behavioral patterns. The proposed technique detects insider threats with an AUC = 0.9047.

In [59], aspect based sentiment analysis and social network information of the user are used to detect insider threat. A potential advantage of aspect based sentiment analysis is that it gives more concrete information about the staff. The proposed approach uses a combination of dl techniques such as Gated Recurrent Unit (GRU) and skipgram for sentiment profile building of users. The user's sentiments are then ranked based on the anomaly score. the data set used was publically available email dataset ENRON. Any existing expert and intelligent systems can be complemented by proposed approach for better results.

In [60] Markov chain model is used to list the user's behavior over time. Sequential data sets were originated according to the impact of n occurrences of Markov attribute and classified by ML algorithm. Only 15% of the CMU dataset was used for evaluating the model and the result was 97% accuracy. Thus the proposed approach shows the effectiveness of Markov chain model for insider threat detection.

2.9.2 GRAPH BASED INSIDER ANOMALY DETECTION

In [49], A graph based insider detection technique is proposed. The proposed technique detects the malicious conduct of an employees based on not only its own activities but also the malicious activities of the employees with same job roles. At first a relationship graph between company's employees is generated using a machine learning technique. Then, prospective insiders are recognized by designing a graph signal processing technique. The insider detection and false positive rates are progressive as compared to the detection rates on individual employees. The proposed technique exhibit that associated behavior of employees inside an organization gives better identification of malicious behavior. The dataset used is CMU CERT V6.2.

In [54] a graph based technique to detect employee's malicious behavior is proposed. Rich contextual data is handled easily using graph based approach and helps us to identify patterns in company's records which are otherwise are difficult to found using statistical techniques or conventional queries. Our approach works by first reporting the normal data usage of the employees, and then identifies any anomalous data patterns related to previously discovered data patterns. Enterprise graph database Neo4j is used for analyzing and visualizing anomalous patterns. Tools like GBAD and Neo4j are used to discover and visualize malicious employees and their communication patterns. Calls, emails, procurement and meetings datasets of employees are analyzed of Kasios, a furniture building company.

In [40] the researcher proposed an insider attack detection mechanism using Gaussian Mixture model. The proposed scheme integrates security experts knowledge as an important system component for reducing the number of false positives which are common in insider-threat detection. The scheme also proposes a new approach in which non-technical indicators of insider threat are included as key elements of the system. A feasibility study is also done at the end to show the effectiveness of the proposed scheme. Each user is represented as 10 dimensional vector. The selected features include PC, login, login after hours etc. The dataset on which the experiment is performed is publically available CMU-CERT dataset V4.2. Metrics upon which results are evaluated are precision, false positive and recall, and

produced results show high precision, low false positives and high recall. However, the proposed scheme looks complex and time intensive. The problem with using GMM is that for computational reasons it often fails if the dimensionality of the problem is too high, which in this scheme is 10 (considerably high). The number of mixture models must be set by user to efficiently fit to the training set, otherwise experiment needs to be done for different number of models to find the best.

In [42] the researcher proposed a hybrid framework consisting of graph analysis and anomaly detection schemes for insider threat detection. The approach consists of two modules "Graphical Processing Unit" (GPU) and "Anomaly Detection Unit" (ADU). Heterogeneous enterprise data (event logs, email logs, http logs etc.) is fed to the GPU which shows the interrelation between network assets by creating a graph. GPU then creates several sub-graphs for each user. Calculated graphs and subgraphs are then fed to the ADU which used "Isolation Forest" algorithm is used for isolating anomalous users by assigning anomaly score, which is used for identifying the user as malicious or normal. The isolation forest algorithm used is light weight and works well with huge data set and high dimensions. It has small memory requirement and linear time complexity. The dataset on which the experiment is performed is highlighted which is publically available CMU-CERT dataset V4.2. 79% user are considered as normal while the remaining are considered malicious. By increasing the no. of input parameters the results can be improved. The scheme does not take into account the social behavior, content analysis of emails and web browsing of users which would otherwise improve the results.

In [64] attributed graphs for showing high dimensional, diverse data for detecting of insider threats is proposed. The combined techniques of attributed graph clustering approaches and outlier ranking in subspaces of attributed graphs is used in the proposed framework. The framework is claimed to be the first one to detect insider threats using attribute graph clustering and outlier ranking approach. Two main subspace/subgraph clustering algorithms used are "EDCAR" and "GAMER" for community detection in attributed graphs. The outlier ranking mechanism "GOutRank" is used.

In [**?**] a framework which uses the combined approach of Structural Anomaly Detection (SA) and Psychological Profiling (PP) of users for insider threat detection is proposed. The techniques used in SA includes graph analysis, dynamic tracing, and ml in order to identify

structural anomalies in substantial information network data, while PP uses behavioral information of individual users to construct dynamic psychological profiles. Threats are identified by the combined outcomes of both the techniques. The proposed approach evaluated on a large data set generated form a multi-player online game, World of Warcraft (WoW) which contains the behaviors of around 350,000 characters over the period of 6 months. SA is used to forecast whether and when characters leave their league. PP approximates the five-factor personality model for all characters. The evaluation results produced by both approaches on gaming dataset are good thus proving the framework to be useful.

2.10 INSIDER ATTACK DETECTION USING OTHER TECHNIQUES

In [44] the researcher proposed a network based insider attack flexible approach "Gargoyle". The trustworthiness of the context of an access request is evaluated through a new set of contextual attributes called Network Context Attribute (NCA), information such as the user's device capacity, security-level, existing and previous interactions with other devices, network connection status, and suspicious online activities are obtained from network traffic analysis. The proposed scheme will produce more efficient results if integrated with recent machine learning techniques.

In [45] the researcher proposed a network packet inspection based insider threat detection scheme. At first network packets (HTTP, FTP, TLS, DNS) are captured and arranged by their sequence number, then their type is accessed by analyzing the content of the packets and at last a report is generated depending on the activity performed. A graph based approach is used to evaluate this scheme, in which a weight is assigned to each activity. Time during which the activities are performed add more value to the weight. Wireshark is used for packet capturing at proxy server.

In [51] the approach used by the researchers is deploying honey pot sensors for insider threat detection. In order to identify the actions of an insider within the company's local network honey pot sensors are used. The categorization of data obtained by honey pot sensor is processed by utilizing the Insider Threat rate classifier in Kabana toolbox for activity identification. Use of honey pot sensor for keeping track of system calls in real time is cost

effective, because there is no need to monitor the system calls from every system. The proposed framework produced accurate results in terms of false positive rate and classification. Insider Threat value is calculated using zero mean error with real time precision.

In [52] a framework using complex event identification is proposed. The model works by properly designing rules and regulations into complex events and impressively examining either employees conduct conforms to the rules and regulations. Reasonable, effective, complete and utter rules and regulations will help in effectively detecting abnormal behavior and threats associated with employees. However, this technique very much relies on the completeness of the rules. In practical scenario, the rules generally have ambiguity. This is also taken into account and employees' daily behavior as common sense is added to enhance the capability to identify internal threats. On the other hand, this method is essentially authoritarian compliance and unable to detect unknown threats.

In [61] in order to predict and detect insider threat, disturbing psychological patterns of individual users are obtained by analyzing electronic communications. For this purpose, multiple text analysis methods which includes lexicon-based emotion analysis, LSTM sentiment classification, SVM emotion classification, and LDA topic modeling are used to form a hybrid psycholinguistic framework. Use of various text analysis methods for psycholinguistic analysis in insider threat avoidance is claimed to be the first study of the sort. Performance of the text analyzers used in the proposed framework achieves acceptable performance.

In [63] a state machine system is proposed that can efficiently integrate policies from rule-based systems and notifications from anomaly detection systems in order to create attack models followed by the insiders to launch an attack. The proposed system helps the security experts to analyze and detect attack patterns by providing a visual interface. It uses the theoretical behavioral knowledge, examine different types of logs for attack graph creation, notifies when an attack pattern is complete producing outputs statistical data as output and shows employees behavior on real time visually. The proposed approach is effective in detecting insider attacks with minimal time and memory requirements. The framework is evaluated on ten scenarios and it was able to detect the perpetrators in seven of these scenarios with no false positives. The system is capable to produce real time alerts.

2.11 CONCLUSION

The comparison shown in the table gives a clear picture of various ML and DL techniques used for Insider Threat detection. Some of the techniques are doing the job efficiently, but have some lacking in terms of complexity, missing performance evaluation metrics. Some models are not evaluated on real life scenarios, and are processing and memory intensive. Some does not clarify the dataset used for evaluation. Some have relatively small test data, which doesn't fully evaluate the performance of the technique. While some hybrid DL techniques are used which combines the advantages of both the approaches, however it also imposes some limitations in terms of processing, memory and dataset requirements. Some techniques on the other hand failed to give an idea of how it contributes to insider threat detection, they are vaguely written and gives no clear idea to the reader.

It is also observed that most Role Based or Behavior Based techniques produces significant quantitative results as compared to graph based and other techniques. The most widely used technique is LSTM ([34, 39, 48, 43, 58, 61]) as it is very effective in natural language processing. It can automatically learn features. It is very good in processing sequence and time series data due to which it is used to model user behavior. Another widely used technique is Deep AutoEncoders, it has the ability to be used on real valued datasets and are quick & concise. Keeping in view of the above discussion, a novel hybrid Deep Learning approach will be designed to detect insiders efficiently, with low processing and memory requirements, with low false positive rate and higher accuracy.

Table 2.1: Behavior Based Insider Attack Detection Techniques

Sr.	Paper Reference	Technique used	Remarks
1	[33]	Supervised Deep Auto Encoder	This paper is an improvement to some-existing techniques, and combines the advantages of both supervised and unsupervised learning. Dataset used is not mentioned.
2	[34]	LSTM-CNN	Experimental setup is not explained, the platform they used for evaluation of results, the memory and processing requirements, the time of training and testing the model are all missing. Dataset used is CMU-CERT dataset V4.2.
3	[35]	XGBoost	The dataset used is CMU-CERT dataset V6.2. The paper claims to be the first one to use XGBoost algorithm for insider threat detection. Achieves 100% recall, and produces good results as compared to random forest, multilayer perceptron and SVM.
4	[37]	Random Forest	The evaluation of the proposed model is performed on both public (CMU-CERT) and private datasets (Next-Labs). The algorithm claims to produce 97.81% accuracy and only 2.19% error.
5	[38]	Dictionary method & CNN	The dataset used is CMU-CERT dataset V4.2. Sentiment classification accuracy of 100% for http and 96% for email content is achieved. This model claims accurate prediction of malicious insider based on their attack motivation predictor.
6	[39]	LSTM	The dataset used is CMU-CERT public dataset. Version of the dataset is not mentioned.The model is trained and tested on relatively small datasets, which doesn't fully depict proposed model effectiveness.
7	[41]	Distance Measurement Techniques (DL Distance, Jaccard Distance and Cosine Distance)	The model is evaluated on synthetic CERT dataset. Real life scenario evaluation of the model is missing. DM techniques only compare previous week user activity with the current week, which creates a high significance of false positives.
8	[48]	LSTM	The dataset used is CMU-CERT public dataset. Version of the dataset is not mentioned. The framework claims to be more efficient than other log based systems and works on real time systems. Proposed approach's comparison with other latest techniques is missing.
9	[43]	Kernal PCA & LSTM-RNN	The dataset used is CMU-CERT dataset V6.2. over tensor flow framework. The proposed scheme produces good results with a precision=95.12% and accuracy=93.85% as compare to isolation forest, SVM and PCA.
10	[46]	Deep AutoEncoder	The dataset used is CMU-CERT V6.2. The platform they used for evaluation of results, the memory and processing requirements, the time of training and testing the model are all missing
11	[47]	Deep Belief Network & One-Class SVM	User characteristics and domain understanding is not required, and the technique is purely data driven. The results in terms of AUC, accuracy, and other performance evaluation scale is missing.The training and testing dataset upon which testing and evaluation is performed is also not mentioned.
12	[58]	Multi State LSTM & CNN	The proposed technique detects insider threats with an AUC = 0.9047. The platform used for evaluation is not mentioned.
13	[59]	Gated Recurrent Unit & Skipgram	The dataset used was publically available email dataset ENRON.The results in terms of AUC, accuracy, and other performance evaluation scale is missing. Proposed approach's comparison with other latest techniques is missing.
14	[53]	Denoising autoencoders, GMM, buck covariance, OCSVM, isolation forest and local outlier factor	TThe dataset used was CMU CERT V4.2. The performance of the technique is compared with an existing technique Pearson-OCSVM and evaluated in terms of accuracy and false positive rate.Achieved accuracy is 87.79%.
15	[60]	Markov Chain Model	Only 15% of the CMU dataset was used for evaluating the model and the result was 97% accuracy. The test data is relatively small.

Table 2.2: Graph based Insider Anomaly detection

Sr.	Paper Reference	Technique used	Remarks
1	[40]	Gaussian Mixture Model	The dataset is publically available CMU-CERT dataset V4.2. Produced results show high precision, low false positives and high recall. Proposed scheme looks complex and time intensive. GMM often fails if the dimensionality of the problem is too high.
2	[42]	Isolation Forest	The dataset used is CMU-CERT V4.2.79% user are considered as normal while the remaining are considered malicious. Results can be improved by increasing no. of input parameter. The scheme does not take into account the social behavior, content analysis of emails and web browsing of users which would otherwise improve the results.
3	[54]	GBAD & Neo4j	Dataset used is of Kasios, a furniture building company.Enterprise graph database Neo4j is used for analyzing and visualizing anomalous patterns.Performance evaluation metrics are missing.
4	[64]	EDCAR & GAMER for attribute graph clustering.GOutRank for outlier ranking	The approach used is complex and difficult to understand. No proper idea can be developed from this research.Performance evaluation metrics are missing.
5	[?]	Structural Anomaly Detection (SA) and Psychological Profiling (PP)	Dataset used is generated from a multi-player online game, World of Warcraft (WoW). The evaluation results produced by both approaches on gaming dataset are good thus proving the framework to be useful.

Table 2.3: Insider threat detection using Other techniques

Sr.	Paper Reference	Technique used	Remarks
1	[44]	Network Context Attribute (NCA), SDN & Function Based Access Control (FBAC)	The first solution to evaluate the context of an access request using network-traffic extracted. The proposed scheme will produce more efficient results if integrated with recent machine learning techniques.
2	[45]	Network Packet Inspection	A graph based approach is used to evaluate this scheme, in which a weight is assigned to each activity. Wireshark is used for packet capturing at proxy server.
3	[51]	Honeypot sensors	Use of honey pot sensor for keeping track of system calls in real time is cost effective. Framework produced accurate results in terms of false positive rate and classification. Insider Threat value is calculated using zero mean error with real time precision.
4	[52]	Complex Event Identification	This method very much relies on the completeness of the rules and regulations. In practical scenario, the rules and regulations generally have ambiguity. This approach is unable to detect unknown threats.
5	[61]	Hybrid psycholinguistic framework (lexicon-based emotion analysis, LSTM sentiment classification, SVM emotion classification, and LDA topic modeling)	Use of various text analysis methods for psycholinguistic analysis in insider threat avoidance is claimed to be the first study of the sort.Performance of the text analyzers achieves acceptable performance.
6	[63]	State Machine System	The proposed approach is effective with minimal time and memory requirements. The framework is evaluated on ten scenarios and it was able to detect the perpetrators in seven of these scenarios with no false positives. The system is capable to produce real time alerts.

Table 2.4: Comaprison between different ML DL techniques

Sr.	Paper Ref.	Approach			Technique			Algorithm	Performance Matrices					Dataset Used		
		Behavior Based	Graph Based	Other	Machine Learning	Deep Learning	Hybrid		Accuracy	Precision	Fscore	AUC	Graph	CMU-CERT V4.2	CMU-CERT V6.2	Other
1	[35]	✓	✗	✗	✗	✗	✓	XGBoost	✓	✗	✓	✗	✗	✗	✓	✗
2	[34]	✓	✗	✗	✗	✗		LSTM-CNN	✗	✗	✗	✓	✗	✓	✗	✗
3	[37]	✓	✗	✗	✓	✗	✗	Random Forest	✓	✗	✗	✗	✗	✓	✗	✓
4	[38]	✓	✗	✗	✗	✗	✓	Dictionary method & CNN	✓	✗	✗	✗	✗	✓	✗	✗
5	[43]	✓	✗	✗	✗	✓		LSTM-RNN	✓	✓	✗	✗	✗	✗	✓	✗
6	[47]	✓	✗	✗	✓	✗	✗	One-Class SVM	✓	✗	✗	✗	✗	✓	✗	✗
7	[46]	✓	✗	✗	✗	✓	✗	Deep Autoencoder	✗	✗	✗	✗	✗	✗	✓	✗
8	[53]	✓	✗	✗	✗	✗	✓	Denoising autoencoders, GMM	✗	✗	✗	✗	✗	✗	✗	✗
9	[41]	✓	✗	✗	✗	✗	✓	Distance Measurement Techniques	✗	✗	✗	✗	✗	✗	✗	✗
10	[58]	✓	✗	✗	✗	✓		Multi State LSTM & CNN	✗	✗	✗	✓	✗	✗	✗	✗
11	[59]	✓	✗	✗	✗	✗	✓	Gated Recurrent Unit & Skipgram	✗		✗	✓	✗	✗	✗	✓
12	[60]	✓	✗	✗	✓	✗		Markov chain model	✓	✗	✗	✗	✗	✗	✗	✗
13	[40]	✗	✓	✗	✓	✗	✗	GMM	✓	✓	✓	✗	✗	✓	✗	✗
14	[42]		✓	✗	✓	✗	✗	Isolation Forest	✓	✗	✗	✗	✗	✓	✗	✗
15	[54]		✓	✗	✗	✗	✓	GBAD & Neo4j	✗	✗	✗	✗	✓	✗	✗	✓
16	[64]	✗	✓	✗	✗	✗	✓	EDCAR & GAMER	✗	✗	✗	✗	✓	✗	✗	✗
17	[?]	✗	✓	✗	✗	✗	✓	Structural Anomaly Detection (SA) and Psychological Profiling (PP)	✗	✗	✗	✗	✓	✗	✗	✓
18	[45]	✗	✗	✓	N/A			Network Packet Inspection	✗	✗	✗	✗	✓	✗	✗	✓
19	[51]	✗	✗	✓	N/A			Honeypot sensors	✗	✓	✓	✗	✗	✗	✗	✓
20	[63]	✗	✗	✓	✗	✗	✓	State Machine System	✗	✗	✓	✗	✓	✗	✗	✗

Chapter 3

MACHINE LEARNING METHODOLOGY AND ITS IMPLEMENTATION

3.1 PROPOSED DEEP LEARNING APPROACH

Insider threat dataset has a multivariate time-series data which consists of several variables discovered over a time interval. On this multivariate time-series data for insider attack detection an **"LSTM Autoencoder"** will be build.

3.2 COMPONENTS OF OUR PROPOSED APPROACH

Our system consists of following components:

3.2.1 Log Assembling

This step consists of gathering data from various sources. The collected data consists of log files of organizational employees generated from various activites. The collected data is then assembles into a centralized repository for further processing.

3.2.2 Log Parsing

At this step the data is processed and is made readable for the proposed deep learning algorithm. Most of the collected data consists of text strings, which makes it incompatible with most of the ml and dl algorithms. So the data is transformed by encoding, the technique will be explained later.

3.2.3 Feature Selection

At this stage the data that will be used to train the deep learning algorithm is extracted. The stage is crucial because the important and mandatory features should not be missed as it will lead to poor performance and faulty detection while on the other hand unimportant, unneccesary feature selection will result in poor prediction. Different user activities, the day and time of activities seems important and necessary in our dataset, as they reveal the behavior of the user more clearly.

3.2.4 Model Training

At this stage the selected model is trained, the machine builds its knowledge base for future predictions either an instance is malicious or benign. The performance of the classifier is evaluated by running the algorithm multiple times and tuning the parameters. The output is evaluated based on performance measures mentioned above in 3.3

3.2.5 Testing

Once the model is trained and produced satisfying results on training data, the algorithm is tested on unseen data to see if it worked as predicted.

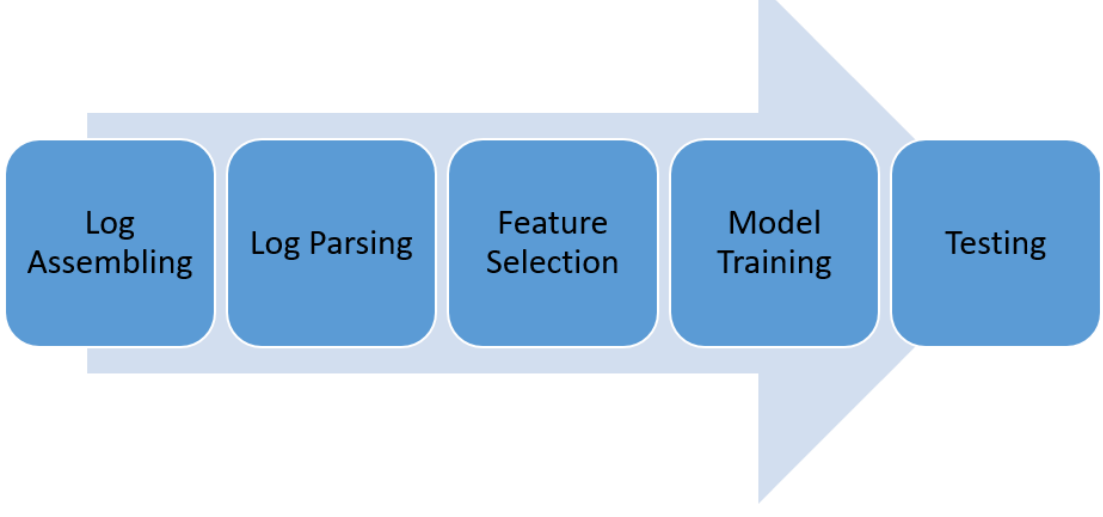

Figure 3.1: Components of our proposed approach

3.3 PERFORMANCE EVALUATION SCALES

3.3.1 Confusion Matrix

A classifier's perfromance is estimated by a confusion matrix. An actual class is represented by a row in confusion matrix, in our case a normal or insider, while each column shows the predicted values. It helps us to understand how a classifier is performing and gives some useful information.

Table 3.1: Confusion Matrix

True Class	**Predicted Class**	
Normal	True Positive (TP)	False Positive(FP)
Insider	False Negative (FN)	True Negative (TN)

3.3.2 Accuracy

Accuracy is a one of ways to measure the potential of a classifier and is defined as:

$$Accuracy = (TP+TN)/(TP+TN+FN+FP)$$

The best value of accuracy is 1 while the worst value is 0. To further evaluate the performance of our classifier: sensitivity, specificty, precision and recall are used [68].The table below gives a brief summary of the remaining evaluation scales:

Table 3.2: Performance Evaluation Scales

Scale	**Description**	**Formula**
Sensitivity	It is the ratio between actual positive examples and predicted true positives by the classifier	*Sensitivity=TP/(TP+FN)*
Specificity	Similar to sensitivity but for negative samples	*Specificity=TN/(TN+FP)*
Precision	The ratio of right positive examples to actual positive examples is precision	*Precision=TP/(TP+FP)*
Recall	Recall is similar to sensitivity	*Recall=TP/(TP+FN)*
F1 Score	By combining precision and recall we obtain F1 score	*F1=2*(Precision*Recall)/(P+R)*

3.4 DATA DESCRIPTION

The dataset used is the CMU CERT synthetic insider threat dataset r4.2. The dataset consists of synthetic data of both normal and malicious insiders. The dataset consists of 1000 synthetic users out of 70 are malicious insiders. The dataset consists of various csv files which includes [70]:

Table 3.3: Csv Files Details

File	**Description**
logon.csv	Log of users logging in and out on a computer
device.csv	Log of users connecting and disconnecting of external devices (USB)
http.csv	Users browser history
email.csv	Email logs
file.csv	Log of users activity on files (coping file to an external device)
psychometric.csv	Contains users personality attributes [69] i-e. Openness: creative/curiois vs careful Conscientiousness: efficient vs irresponsible Extraversion: passionate vs shy Agreeableness: friendly vs disconnected Neuroticism: nervous vs confidents
LDAP (Lightweight Directory Access Protocol)	Set of files describing all users and their assigned job roles.

Table 3.4: Csv Files Data details

File	Features	Data Types
Logon.csv	Id	string
	date	string
	user	string
	pc	string
	activity	string
device.csv	Id	string
	date	string
	user	string
	pc	string
	activity	string
http.csv	Id	string
	date	string
	user	string
	pc	string
	url	string
	content	string
email.csv	Id	string
	date	string
	user	string
	pc	string
	to	string
	cc	string
	bcc	string
	from	string
	size	integer
	attachment_count	integer
	content	string
file.csv	Id	string
	date	string
	user	string
	pc	string
	filename	string
	content	string
psychometric.csv	employee_name	string
	user_id	string
	O	integer
	C	integer
	E	integer
	A	integer
	N	integer
LDAP.csv	employee name	string
	user id	string
	email	string
	role	string
	projects	string
	business_unit	string
	functional_unit	string
	department	string
	team	string
	supervisor	string

3.5 INSIDER THREAT SCENARIOS

In this dataset, malicious insider user are designed to accomplish one out of the following two scenarios at some point in time.

1. Use of external hard drives, or work after hours, login activity after office hours by the user who did not have such previous routine, using of a flash drive, and uploading data to wikileaks.org and then leaving the organization shortly thereafter.

2. User visiting job sites and seeking employment from a competitor. Before leaving the organization, they use a flash drive (at markedly higher rates than their previous activity) to steal data.

The reason for selecting version r4.2 is that most datasets had one instance of each scenario. Dataset 4.2 was a "dense needle" dataset and had many instances of each scenario.

3.6 DATA PRE-PROCESSING

3.6.1 Dataset

The dataset contains multiple csv files as described in 3.4. To make the approach simple all the csv files are aggregated and most relevant features are extracted.
Total number of rows are 32,770,220.

3.6.2 Features

Psychometric.csv is not used in feature selection. Features from all other csv files included integer encoded day, time, pc, user_id, user_role,user_functiona_unit, user_department and activity features. Id of the features is redundant and is not included.
Possible feature values are shown in the table below:

Table 3.5: Feature Values

Features	**Values**
Day	0-6
Time	1-24
Activity	1-7
User_id	1-1000
User_role	1-42
User_functional_unit	1-6
User_department	1-7
PC	Unique number

Activity are labeled as follows':

Table 3.6: Activity Labels

Activity	**Label**
Logon	1
Logoff	2
Connect	3
Disconnect	4
E-mail	5
File	6
Http	7

Days are labeled as':

Table 3.7: Days Labels

Days	**Label**
Monday	0
Tuesday	1
Wednesday	2
Thursday	3
Friday	4
Saturday	5
Sunday	6

Each user has an assigned a role inside the organization. The encoding scheme of User Functional Unit is as follows:

Table 3.8: User_Functinal_unit Encoding

User_Functinal_Unit	**Label**
Administration	1
Research And Engineering	2
Manufacturing	3
Finance	4
Sales And Marketing	5
Purchasing And Contracts	6

3.6.3 Encoding

The collected features from various data sources contains multiple categorical and ordinal values given as text strings,and will not be used as input for these algorithms. Therefore,

the values will have to be suitably encoded, for the algorithms to make correct prediction. Presence of a feature is represented by 1 while the absence by 0.

3.7 PLATFORM USED FOR IMPLEMENTATION

3.7.1 Programming Language

The proposed framework is developed using Anaconda which is a free and open-source distribution of python and R programming. the programming language used is Python 3.7.1. the popularity of python is increasing everyday because it is an open source high-level programming language and it supports a large number of practical tools for ml and dl applications [71].

3.7.2 Libraries

- **TensorFlow** A high performance open source, end-to-end python library for efficient and high speed numerical calculation allowing users to create high level deep learning and ML applications. It allows us to create and train easy ML models with high level APIs like Keras. TensorFlow has gained incredible growth and acceptance in the data science community [72]. The version used is 1.13.1.

- **Pandas** An open source BSD-licensed library providing high level performance, with simple and easy-to-use data structures, and data exploration tools [73].

- **Numpy** In order to support large, multidimensional arrays and matrices, numpy is the perfect python library. A wide range of sophisticated mathematical functions is available to operate on these arrays [74].

- **Scikit-learn** Free and open source python library for machine learning that that provides many unsupervised and supervised learning algorithms and supports interoperability with other python libraries [28].

- **Seaborn** Matplotlib based python data imaging library. It provides an outstanding drawing interface and informative statistical graphics. [75].

- **Keras** A high-level python neural network API, runs on top of TensorFlow, CNTK, or Theano. Both conventional and RNN on CPU and GPU are supported. Allows fast and easy modeling [76].

3.7.3 Platform

Anaconda: World renowned data science platform that serves as the core for modern ml. It carry out data science and ml tasks at speed and scale, unveiling the full capability of data science and ml initiatives [77]. The version used: 2018.12, Build: py37_0.

3.8 CHALLENGES

3.8.1 Dataset Selection

One of the biggest challenge is to find a good dataset. Real world insider threat datasets are not publically available. Due to privacy issues it was not possible to gather data from any organization. Also it was not possible to maintain complete anonymity of personal data. Due to these issues, publically available CMU CERT synthetic dataset was used. Now a days most of the research on insider threat is carried out on this dataset. Also these are no privacy constraints on this synthetic dataset.

3.8.2 Feature Selection

Selecting features and purifying the dataset is an iterative approach, that is to repeated at regular intervals in order to improve the results. Envisioning how the machine would interpret the data that we provided has proved to be challenging. This become more obvious when we use synthetic datasets with very detailed scenarios.

3.8.3 Large Dataset

Initially, we worked on small portions of the total dataset. With the help of this sampling we were able to identify the effectiveness of our proposed approach and help us to parse the information correctly. But it became difficult for us to gather the results produced from the whole dataset. Processing such large dataset faces some challenges like processing power , internal memory and storage capacity. So we sampled small portions of data at first to identify the processing requirements. Over all the system is processing intensive, but at only few steps in the model.

3.8.4 Missing Data

There were some missing data in this synthetic dataset to make itlook like real life data collected from sensors. The algorithms cannot work with this missing data. In order to cope with this problem,the simplest solution was to remove all such instances, but it would result

in loosing lot of important information and insiders tracts would be lost. Another approach would be to replace the missing values with a default value such as 0 or -1 to, however it would be easy for us but these values would be perceived as actual data points by the algorithm which would change the results. So it is decided to replace the missing values with the estimated mean value of that feature.

Chapter 4

PROPOSED SOLUTION AND EXPERIMETAL RESULTS

4.1 INTRODUCTION

An experimental setup was established to evaluate and study the importance and usefulness of the proposed technique. The experimental environment consists of AMD A8 pro 1.9 GHz CPU,8 GB RAM, Windows 10 Home. The testing is carried out on Anaconda 2018.12, Build: py37_0 using Jupyter Notebook 5.7.4, which is a webbased, interactive programming environment enabling user to run and edit human readable documents.

4.2 PROPOSED STRUCTURE

Step 1: The very 1st step was the data pre-processing, which includes feture encoding and coping up with the issue of missing values. As the dataset is very large and processing it manually would not be practical, so a code is written to automate the process.

```
Start
Import Libraries (Pandas, Numpy, datetime)
Input: Read_csv_files (device.csv, emails.csv, files.csv, https.csv,
logon.csv)
Use columns = date, user, pc, activity
Dataset = concat (device, emails, files, https, logon)
Convert activity column to integer
Split datetime into date and time
Input: Read all LDAP files and concatenate them.
Dictionary creation and assign each user a unique integer value
Value_id=userid[*][*]
Insert Insiders
Known insiders are marked
Key= User_id
if  key in insiders
Dict_insider[key]= [1]
else
```

```
Dict_insider[key]= [0]


Fill empty spaces = mean values
Create new Master Dataset
END
```

Step 2: After this the data is prepared to be fed to the proposed algorithm.LSTM takes a 3D array as input.

samples×*lookback*×*features*

- *samples*: simply the data points
- *lookback*: Process data up to (t-lookback) to make a prediction at a given time t.
- *features*: No. of features.

```
Split data = T_data, V_data, Tst_data
```

Step 3: Once the data is ready, it is divided into train, valid and test data. This is done by using sklearn function. The X is then transformed into the required 3D array *samples*×*lookback*×*features*

```
Start
Autoencoder is trained with negative label data i-e label=0
T_data = T_data[label=0]
T_data= reshape(t_data.shape[*], loockback, n_features)
End
```

Step 4: LSTM-Autoencoder Training

At first a few varibles are intialized and then a simple LSTM-Autoencoder architecture is developed.

```
Initialize Variables
Epochs =200, batch=64, learning_rate=0.0001
Encoder is constructed
LSTM_autoEncode.add (LSTM (32, active=    relu    ,
inp_shape, return_seq=   T   ))


Decoder is constructed
```

```
LSTM_autoEncode.add (LSTM (16, active, return_seq=  T  ))
```

```
Colocations handled automatically by placer.
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 20, 32)            5376
_________________________________________________________________
lstm_2 (LSTM)                (None, 16)                3136
_________________________________________________________________
repeat_vector_1 (RepeatVecto (None, 20, 16)            0
_________________________________________________________________
lstm_3 (LSTM)                (None, 20, 16)            2112
_________________________________________________________________
lstm_4 (LSTM)                (None, 20, 32)            6272
_________________________________________________________________
time_distributed_1 (TimeDist (None, 20, 9)             297
=================================================================
Total params: 17,193
Trainable params: 17,193
Non-trainable params: 0
```

Figure 4.1: LSTM-Autoencoder Training

Next the Autoencoder will be trained.

```
adam = optim.Adam(lr)Start
Optim= adam(L_rate)
Compile Loss
LSTM_autoEncode.fit (T_data, epochs, batch, V_data(V_data))
```

```
Train on 591302 samples, validate on 147767 samples
Epoch 1/200
 - 696s - loss: 0.6402 - val_loss: 0.5316
Epoch 2/200
 - 618s - loss: 0.5123 - val_loss: 0.5015
Epoch 3/200
 - 602s - loss: 0.4938 - val_loss: 0.4880
Epoch 4/200
 - 578s - loss: 0.4820 - val_loss: 0.4785
Epoch 5/200
 - 562s - loss: 0.4754 - val_loss: 0.4736
Epoch 6/200
 - 686s - loss: 0.4695 - val_loss: 0.4669
```

Figure 4.2: Autoencoder Training

Loss over the epochs is plotted

```
Plot= (lstm_autoEncode[loss], label=Train)

Plot.show()
```

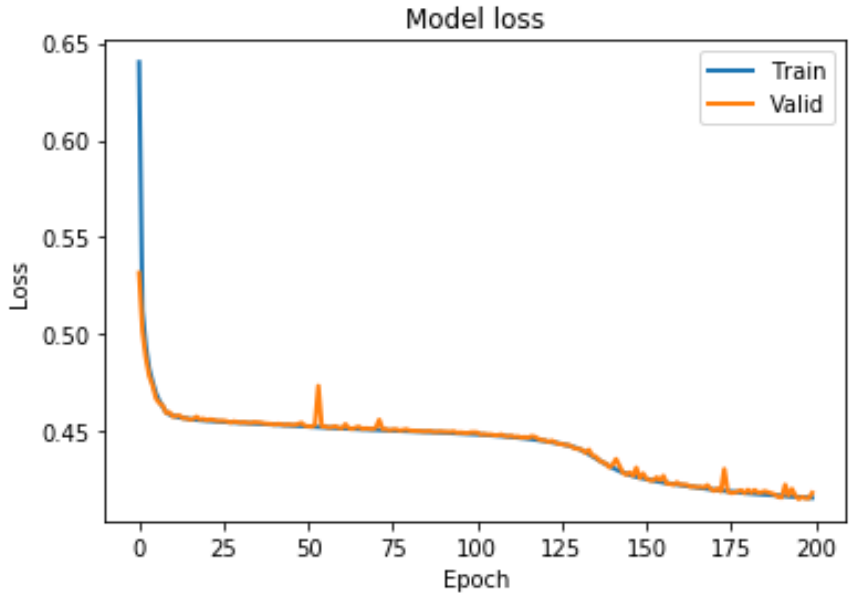

Figure 4.3: Model Loss

4.3 EXPERIMENTAL RESULTS

Classification

```
Predict= LSTM_autoEncode .predict(Valid_data)
Precision_recall_curve=Error_def.TrueClass,Error_def.Reconstruct_Error
Plot = Precision & Recall for different threshold values
Plot.show()
```

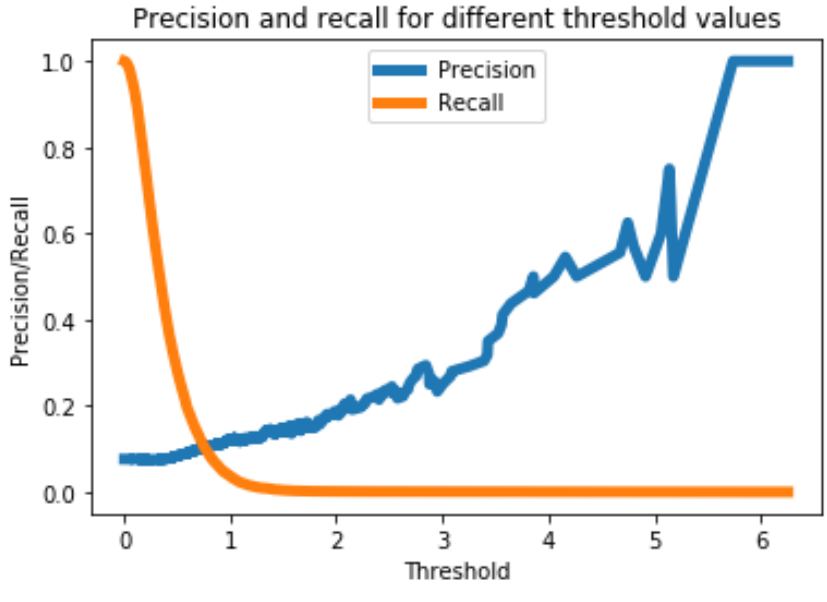

Figure 4.4: Precision and Recall for threshold values

The classification will now be tested or predicted on the test data.

```
Predict =LSTM_autoEncode .predict(Test_data)
Threshold =1
Plot = Reconstruct Error for different classes
Plot.show()
```

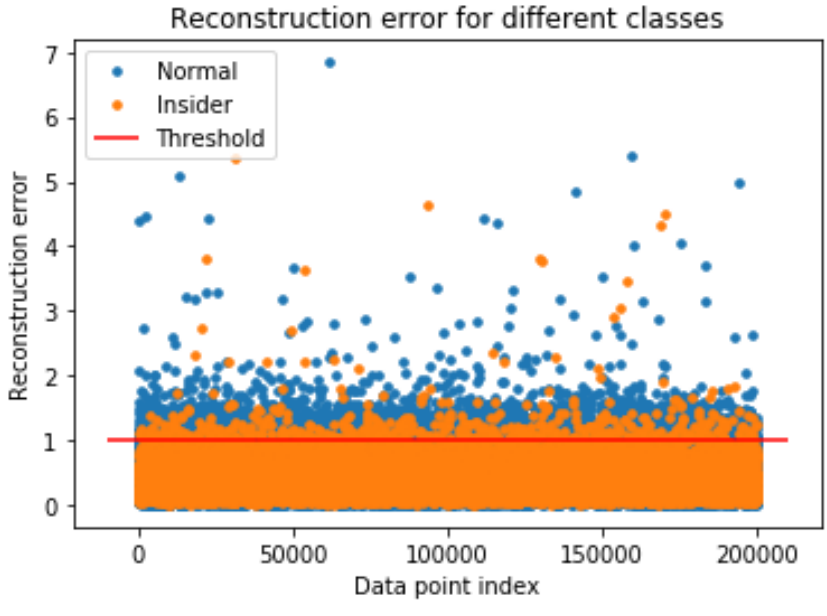

Figure 4.5: Reconstruction error for different classes

Step 6: Test Accuracy Confusion Matrix

```
Confusion Matrix = Confusion_matrix(error_df, True_class, predictedClass)
Plot.show()
```

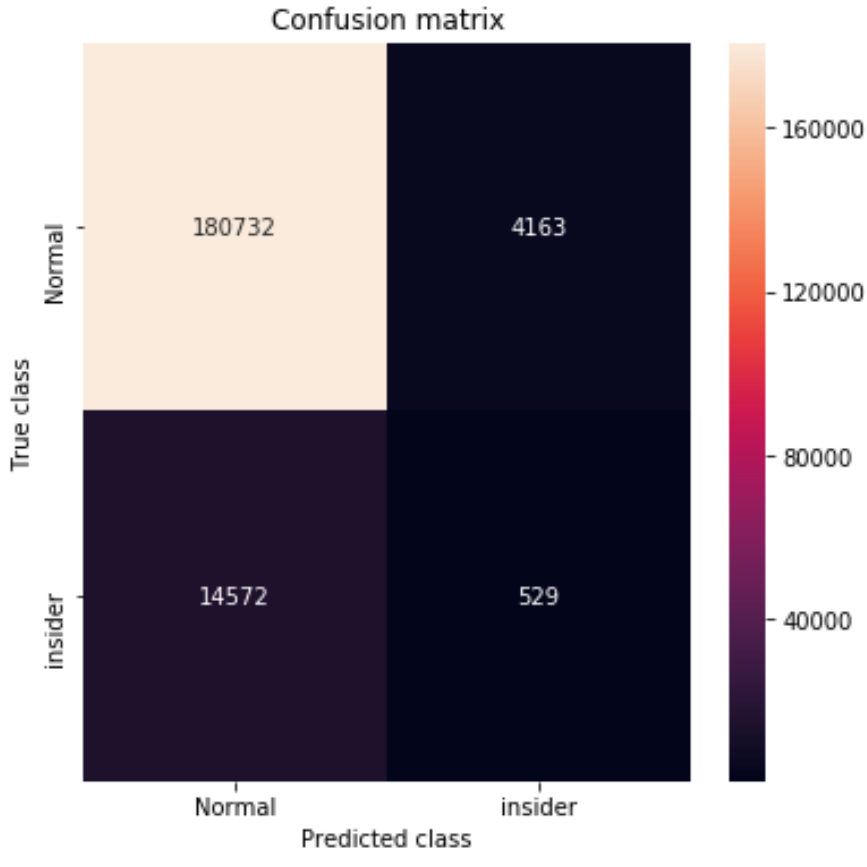

Figure 4.6: Confusion Matrix: shows the TP and FP

4.4 RESULTS AND DISCUSSION

The dataset used is the CMU CERT synthetic dataset r4.2 which consists of 1000 synthetic users out of which 70 are malicious insiders.The threat scenarios covered are 1) the use of removable drives or work after hours, logging in office computer after hours by the employee which has no such previous history. 2) user start surfing job websites and use of thumb drives at a higher rate(which could be an attempt to steal data or deploy a logic bomb into the system). LSTM is so far considered best in dealing wth timeseries data as it has the ability to remember previous timestamp information and produce more accurate results as compared to other techniques.Autoencoders on the other hand is an unsupervised learning technique. Deep AutoEncoders, has the ability to be used on real valued datasets and are quick & concise. The data we dealt with is a multivariate timeseries data so an LSTM-AutoEncoder is built on this data. Remembering information for long periods of time is practically their default behaviour and hence they have an advantage over normal auto-encoders. So it is one of the best technique for finding anomalies in time series data.

As the dataset used is large so at first the algorithm is trained with a batch size of 32 and epochs=100, but the algorithms performance improves as the batch size and no of epochs increases. So the final model is trained with a batch size of 64 and epochs=200. The results obtained from the above experimentaion is calculated in term of the **Perfromance Evaluation Scales** 3.3 with an **Accuracy** of **90%** and **Precision** of **97%**.The remaining results are shown in the table given below.

Table 4.1: Performance Evaluation

Performance Evaluation Scale	**Values**
Specificity	0.11
Recall	0.92
F1 Score	0.94
FPR	0.9

4.5 COMPARISON OF RESULTS WITH OTHER PROPOSED TECHNIQUES

Our proposed algorithm's performance is compared to other well known techniques i-e. **LSTM-CNN, LSTM-RNN, One Class SVM, Multi State LSTM & CNN,Gated Recurrent Unit & Skipgram**, and upon comparison it is observed that our novel approach produces relatively good **Acuuracy(90.6%), Precision (97%) and F1 Score (94.4%)**.The detailed comparison is shown in the graph below.

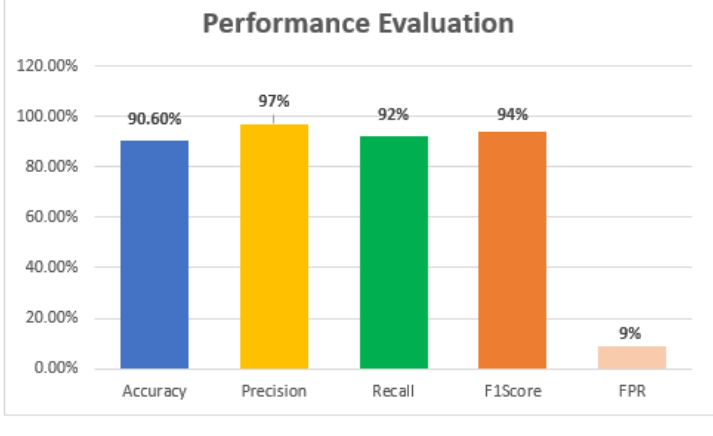

Figure 4.7: Experimental Results

Table 4.2: Comaprison of Results

Sr.	Paper Ref.	Technique Used		Algorithm	Performance Matrices				Dataset Used	Feature Set
		Behavior Based	Graph Based		Accuracy	Precision	Fscore	AUC		
1	[34]	✓	✗	LSTM-CNN	✗	✗	✗	0.94	V4.2	User activity based feature set
2	[43]	✓	✗	LSTM-RNN	93.85%	95.12%	✗	✗	V6.2	Detail is missing
3	[47]	✓	✗	One-Class SVM	87.79%	✗	✗	✗	V4.2	Domain Based Features (logon ,log off time, userId, user_role, functional_unit etc)
4	[58]	✓	✗	Multi State LSTM & CNN	85%	✗	✗	0.9047	V6.2	User Behavior based features (first access time, last access time, activity etc)
5	[42]	✗	✓	Isolation Forest	79%	✗	✗	✗	V4.2	Logon/Logoff Events, Psychometric Observations, Web Access Patterns
6	[89]	✓	✗	LSTM based Autoencoder	90%	✗	✗	✗	V4.2	Logon/Logoff activity, file activity, Device activity
7	Proposed Approach	✓	✗	LSTM-AutoEncoder	90%	97%	94%	✗	V4.2	Logon/Logoff Events ,UserId, User_role, Functional_unit, Department, Day, Time ,PC

4.6 CONCLUSION

The proposed technique is evaluated using CMU CERT Dataset V4.2 and it shows promising results in terms of **Accuracy, Precision and F1-Score**. The efficiency of the model will be improved as it learns and trains over time. The problem of over-fittng will be avoided by increasing the size of the data samples, thus improving accuracy with low false positive rate.Our model is well trained, requires minimum doamin knowledge and resources in terms of time and computational complexity.

Chapter 5

CONCLUSION AND FUTUREWORK DIRECTIONS

5.1 CONCLUSION

We study the insider threat problem, and identified that mitigating this problem is a challenging task. Now a days, mitigation against this threat is achieved by implementing user access controls,user behavior monitoring and physical security controls.

In this work a Deep Learning based Insider Attack Detection scheme is presented. The main aim behind the developement of this scheme is its application on user technical data within an organization. Moreover, we want the system to be simple, adaptable and minimum domain knowledge requirement.

The followiing contributions are made to address insider threat problem:

I. Insider threat problem is studied in detail and relevant literature is consulted for an indepth understanding of the problem.

II. The dataset used is the CMU CERT synthetic insider threat dataset r4.2. The dataset consists of synthetic data of both normal and malicious insiders. The dataset consists of 1000 synthetic users out of 70 are malicious insiders.

III. Insider threat scenarios used are:

   a. The use of removable drives or work after hours, logging in office computer after hours by the employee which has no such previous history.

   b. User start surfing job websites and use of thumb drives at a higher rate(which could be an attempt to steal data or deploy a logic bomb into the system).

IV. In our problem, we have a multivariate time-series data. We build an "LSTM Autoencoder" on this data for insider attack detection.

V. An experimental setup was established to evaluate and study the importance and usefulness of the proposed technique.

VI. The testing is carried out on Anaconda 2018.12, Build: py37_0 using Jupyter Notebook 5.7.4, which is a webbased, interactive programming environment enabling user to run and edit human readable documents.

VII. Our proposed algorithm's performance is compared to other well known techniques i.e. LSTM-CNN, Random Forest, LSTM-RNN, One Class SVM, Markov Chain Model,Multi State LSTM & CNN, Gated Recurrent Unit & Skipgram and upon comparison it is observed that our novel approach produces relatively good Acuuracy(90.60%), Precision(97%) and F1 Score (94%)

5.2 ANSWERS OF RESEARCH QUESTIONS

Q1. How can deep learning be utilized for the efficient detection of insider threats with minimum domain knowledge required?

Ans. Deep Learning can be efficiently used to detect insider threats within an organization.The algorithms are capable to analyze organizational data and detect abnormalities which could be helpful in identifying ongoing malicious activity. The results produced can be very helpful in narrowing down the investigaton scope and improves the efficiency of manual investigation. However, it must be stated that we canot solely rely on an automated DL or ML based technique due to the high possibility of false positve rate.

Q2. How can the devised technique produce efficient results with high accuracy and low false positive rate and be applied with minimum resources?

Ans. As this is a DL based scheme, and the efficiency improves as the model learns and trains, performance improves as the model trains over time. It also depends on the data samples, more the size of the data, less will be the chance of over-fitting and higher will be the accuracy with less false positives. So our proposed scheme is well trained, uses a large dataset and produced results with high accuracy(93%) and requires minimum resources to train and test.

5.3 FUTURE DIRECTIONS

In this section, proposals for the future work are provided.

I. The proposed approach is evaluated using CMU-CERT dataset V4.2. Evaluating the approach using newer versions of this dataset will be helpful in identifying the pros and cons of the devised approach and might help in improving the performance. However pre-processing of the newer datasets would be a challenging task.

II. In order to create a robust Insider detection system we need to create more diverse insider threat scenarios, as there is a lack of publically available threat scenarios. This will help us in solving the insider problems with more creativity, high qulaity and accuracy.

III. Data sample size can be increased, thus training the algorithm with increased data samples can improve the results in terms of AUC and ROC values, as there is still a room for improvement in terms of these values.

IV. Features such as email content, URL content etc can be used for sentiment analysis. so it is suggested that such features can be used to extend the proposed system's functionality.

BIBLIOGRAPHY

[1] Insider Report 2018 ca technologies

[2] Mehul S. Raval, Ratnik Gandhi, and Sanjay Chaudhary. Insider Threat Detection: Machine Learning Way Springer Nature Switzerland AG 2018

[3] https://insights.sei.cmu.edu/sei_blog/2017/10/machine-learning-and-insider-threat.html

[4] http://intellspot.com/unsupervised-vs-supervised-learning

[5] https://www.mathworks.com/discovery/deep-learning.html

[6] insider threat report 2019

[7] https://securityintelligence.com/these-5-types-of-insider-threats-could-lead-to-costly-data-breaches/

[8] https://lab.getapp.com/types-of-insider-threats/

[9] http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/

[10] https://www.dni.gov/files/documents/ICA_2017_01.pdf

[11] E. Cole and S. Ring, Insider Threat: Protecting the Enterprise from Sabotage, Spying, and Theft: Protecting the Enterprise from Sabotage, Spying, and Theft. Syngress, 2005.

[12] Duran, Felicia, Stephen H. Conrad, Gregory N. Conrad, David P. Duggan, and Edward Bruce Held. "Building a system for insider security." IEEE Security Privacy 7, no. 6, pp. 30–38, 2009.

[13] https://haystax.com/blog/ebook/insider-attacks-industry-survey/

[14] Herbig, K. "Changes in espionage by Americans 1947–2007," Monterey, CA, Defense Personnel Security Research Center. 2008

[15] Band, S.R., Cappelli, D.M., Fischer, L.F., Moore, A.P., Shaw, E.D. and Trzeciak, R.F., 2006. Comparing insider IT sabotage and espionage: A model-based analysis (No. CMU/SEI-2006TR-026). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

[16] Rost, Johann. “Political reasons for failed software projects” IEEE Software 21, no. 6, pp. 103–104, 2004. 36.

[17] Thompson, Hugh. “The human element of information security” IEEE Security Privacy 11, no. 1 pp. 32–35, 2013.

[18] Theoharidou, M.; Kokolakis, S.; Karyda, M.; Kiountouzis, E. The insider threat to information systems and the effectiveness of ISO17799. Comput. Secur. 2005, 24, 472–484. [CrossRef]

[19] Wong, W.-K.; Moore, A.; Cooper, G.; Wagner, M. Rule-based anomaly pattern detection for detecting disease. In Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02), Edmonton, AB, Canada, 28 July–1 August 2002; pp. 217–223

[20] Eldardiry, H.; Sricharan, K.; Liu, J.; Hanley, J.; Price, B.; Brdiczka, O.; Bart, E. Multi-source fusion for anomaly detection: Using across-domain and across-time peer-group consistency checks. JoWUA 2014, 5, 39–58.

[21] Lunt, T.F,Jagannathan, R, Lee, R, Whitehurst, A Listgarten. Knowledge-based intrusion detection. In Proceedings of the Annual AI Systems in Government Conference, Washington, DC, USA, 27–31 March 1989; pp. 102–107.

[22] Eberle, W.; Graves, J.; Holder, L. Insider threat detection using a graph-based approach. J. Appl. Secur. Res. 2010, 6, 32–81.

[23] Insider Threat Detection Based on User Behavior Modeling and Anomaly Detection Algorithms Junhong Kim, Minsik Park, Haedong Kim, Suhyoun Cho and Pilsung Kang

[24] Mayhew, M.; Atighetchi, M.; Adler, A.; Greenstadt, R. Use of machine learning in big data analytics for insider threat detection. In Proceedings of the MILCOM 2015—2015 IEEE Military Communications Conference, Tampa, FL, USA, 26–28 October 2015; pp. 915–922.

[25] Eric D Shaw, Lynn F Fischer, and Andrée E Rose. 2009. Insider risk evaluation and audit. Technical Report. Defense Personnel Security Research Center Monterey CA.

[26] Marisa Reddy Randazzo, Michelle Keeney, Eileen Kowalski, Dawn M Cappelli, and Andrew P Moore. 2005. Insider threat study: Illicit cyber activity in the banking and finance sector. (2005).

[27] Vijay Vaishnavi and William Kuechler. “Design research in information systems.” (2004).

[28] A Holistic Approach to Insider Threat Detection Sondre Johannessen Berdal

[29] Lockheed Martin Corporation. "https://www.lockheedmartin.com/enus/capabilities/cyber/cyber-kill-chain.html

[30] Dtex Systems. "https://dtexsystems.com/the-insider-threat-killchain- 5-steps-to-watch-out for/".

[31] ZoneFox. "https://www.zonefox.co.uk/media/1305/zf_whitepaper_introducinsgthe-insider-threat-kill-chain.pdf".

[32] Pratik Chattopadhyay, Lipo Wang, and Yap-Peng Tan, “Scenario-Based Insider Threat Detection From Cyber Activities”, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS 2018.

[33] Fangfang Yuan, Yanan Cao, Yanmin Shang,Yanbing Liu, Jianlong Tan, and Binxing Fang “Insider Threat Detection with deep Neural Networks”, Springer International Publications 2018.

[34] Wei Jiang, Yuan Tian, Weixin Liu, and Wenmao Liu “An Insider Threat Detection Method Based on User Behavior Analysis”, IFIP International Federation for Information Processing 2018 Springer Nature Switzerland AG 2018.

[35] Chenchen Liu, Yi Zhong, Yuanli Wang, “Improved Detection of User Malicious Behavior through Log Mining based on IHMM”, IEEE International Conference on Systems and Informatics (ICSAI 2018).

[36] Zahedeh Zamanian, Ali Feizollah , Nor Badrul Anuar, Miss Laiha Binti Mat Kiah, Karanam Srikanth, Sudhindra Kumar, “User Profiling in Anomaly Detection of Authorization Logs”, Springer Nature Singapore Pte Ltd. 2019.

[37] Jianguo Jiang, Jiuming Chen, Kim-Kwang Raymond Choo, Kunying Liu, Chao Liu, Min Yu, Prasant Mohapatra, “Prediction and Detection of Malicious Insiders’ Motivation based on Sentiment Profile on Webpages and Emails”, Milcom 2018 Track 3 - Cyber Security and Trusted Computing IEEE.

[38] Dongxue Zhang, Yang Zheng, Yu Wen, Yujue Xu, Jingchuo Wang, Yang Yu, Dan Meng “Role-based Log Analysis Applying Deep Learning for Insider Threat Detec-

tion", SecArch'18, October 15, 2018, Toronto, ON, Canada 2018 Association for Computing Machinery.

[39] Kholood Al tabash, Jassim Happa, "Insider-Threat Detection using Gaussian Mixture Models and Sensitivity Profiles", Computers Security (2018).

[40] Owen Lo , William J. Buchanan , Paul Griffiths, and Richard Macfarlane, "Distance Measurement Methods for Improved Insider Threat Detection", Hindawi Security and Communication Networks Volume 2018.

[41] Anagi Gamachchi, Li Sun and Serdar Boztas, "A Graph Based Framework for Malicious Insider Threat Detection", 50th Hawaii International Conference on System Sciences (HICSS) 2017.

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

[43] Arash Shaghaghi, Salil S. Kanhere, Mohamed Ali Kaafary, Elisa Bertino and Sanjay Jha. Gargoyle: A Network-based Insider Attack Resilient Framework for Organizations. Association for Computing Machinery 2018

[44] Dinesh Patil, Bandu Meshram. Network Packet Analysis For Detecting Malicious Insider. IEEE 2018

[45] Liu Liu, Olivier De Vel, Chao Chen, Jun Zhang, Yang Xiang "Anomaly-based Insider Threat Detection using Deep Autoencoders" 2018 IEEE International Conference on Data Mining Workshops (ICDMW)

[46] Lingli Lin, Shangping Zhong, Cunmin Jia, Kaizhi Chen "Insider threat detection based on deep belief network feature representation" 2017 International Conference on Green Informatics

[47] Jiuming Lu, Raymond K. Wong "Insider Threat Detection with Long Short-Term Memory" ACSW '19, January 29–31, 2019, Sydney, NSW, Australia

[48] Pratibha, Junshan Wang, Saurabh Aggarwal, Feng Ji, and Wee Peng Tay "Learning Correlation Graph and Anomalous Employee Behavior for Insider Threat Detection" 2018 21st International Conference on Information Fusion (FUSION)

[49] Xuebin Wang, Qingfeng Tan, Jinqiao Shi, Shen Su and Meiqi Wang “Insider Threat Detection Using Characterizing User Behavior” 2018 IEEE Third International Conference on Data Science in Cyberspace

[50] Muhammad Mudassar Yamin(), Basel Katt, Kashif Sattar and Maaz Bin Ahmad .“Implementation of Insider Threat Detection System Using Honeypot Based Sensors and Threat Analytics” Springer Nature Switzerland AG 2020

[51] Zheng Li1 and Kun Liu. “An Event Based Detection of Internal Threat to Information System” Advances in Intelligent Systems and Computing Springer Nature Switzerland

[52] Zhaoyang Zhang, Shen Wang and Guang Lu.“An Internal Threat Detection Model Based on Denoising Autoencoders” Springer Nature Singapore

[53] Sirisha Velampalli, Lenin Mookiah, William Eberle.“Discovering Suspicious Patterns Using a Graph Based Approach” The Thirty-Second International Florida Artificial Intelligence Research Society Conference (FLAIRS-32)

[54] “A Trust Aware Unsupervised Learning Approach for Insider Threat Detection” Maryam Aldairi, Leila Karimi, James Joshi 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science

[55] Junhong Kim, Minsik Park, Haedong Kim, Suhyoun Cho and Pilsung Kang.“Insider Threat Detection Based on User Behavior Modeling and Anomaly Detection Algorithms” Journal of applied science 2019

[56] Gaurang Gavai, Kumar Sricharan, Dave Gunning, John Hanley, Mudita Singhal, and Rob Rolleston.“Supervised and Unsupervised methods to detect Insider Threat from Enterprise Social and Online Activity Data” Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 6

[57] Malvika Singh , B.M.Mehtre, S.Sangeetha .“User Behavior Profiling using Ensemble Approach for Insider Threat Detection” 2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)

[58] Charlie Soh , Sicheng Yu , Annamalai Narayanan, Santhiya Duraisamy, Lihui Che . Employee profiling via aspect-based sentiment and network for insider threats detection Expert Systems With Applications Elsevier 2019

[59] Dong-Wook Kim, Sung-Sam Hong and Myung-Mook Han . “A study on Classification of Insider threat using Markov Chain Model” KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 12, NO. 4, Apr. 2018

[60] Sang-Sang Tan, Santhiya Duraisamy, Jin-Cheon Na . Unified Psycholinguistic Framework: An Unobtrusive Psychological Analysis Approach Towards Insider Threat Prevention and Detection Journal of Information Science Theory and Practice 2019

[61] Schultz, E. E. (2002). A framework for understanding and predicting insider attacks. Computers Security, 21(6), 526-531.

[62] Haozhe Zhang, Ioannis Agrafiotis, Arnau Erola, Sadie Creese, and Michael Goldsmith.A State Machine System for Insider Threat Detection Springer Nature Switzerland AG 2019

[63] Anagi Gamachchi, Serdar Boztas. Insider Threat Detection Through Attributed Graph Clustering 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2017.

[64] Oliver Brdiczka, Juan Liu, Bob Price, Jianqiang Shen, Akshay Patil, Richard Chow, Eugene Bart, Nicolas Ducheneaut. “Proactive Insider Threat Detection through Graph Learning and Psychological Context” IEEE Symposium on Security and Privacy Workshops 2012

[65] Arthur L Samuel. “Some studies in machine learning using the game of checkers.” In: IBM Journal of research and development 3.3 (1959), pp. 210–229.

[66] Aurélien Géron. Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems." O’Reilly Media, Inc.", 2017.

[67] Stephen Marsland. Machine learning: an algorithmic perspective. CRC press, 2015.

[68] https://en.wikipedia.org/wiki/Big_Five_personality_traits

[69] https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099

[70] K Jarrod Millman and Michael Aivazis. “Python for scientists and engineers.” In: Computing in Science Engineering 13.2 (2011), pp. 9–12.

[71] https://www.anaconda.com/tensorflow-in-anaconda/

[72] Wes McKinney et al. “Data structures for statistical computing in python.” In Proceedings of the 9th Python in Science Conference. Vol. 445. Austin, TX. 2010, pp. 51–56.

[73] Travis E Oliphant. A guide to NumPy. Vol. 1. Trelgol Publishing USA, 2006.

[74] https://seaborn.pydata.org/

[75] https://keras.io/

[76] https://www.anaconda.com/about-us/

[77] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with LSTM. (1999).

[78] Chaochun Liu, Huan Sun, Nan Du, Shulong Tan, Hongliang Fei, Wei Fan, Tao Yang, Hao Wu, Yaliang Li, and Chenwei Zhang. 2016. Augmented LSTM Framework to Construct Medical Self-diagnosis Android. In Data Mining (ICDM), 2016 IEEE 16th International Conference on. IEEE, 251–260.

[79] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long short term memory networks for anomaly detection in time series. In Proceedings. Presses universitaires de Louvain, 89.

[80] Haşim Sak, Andrew Senior, and Françoise Beaufays. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association.

[81] https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1 3e5c6f017726

[82] https://www.jeremyjordan.me/autoencoders/

[83] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[84] https://en.wikipedia.org/wiki/Long_short-term_memorys

[85] https://medium.com/datadriveninvestor/deep-learning-different-types-of-autoencoders-41d4fa5f7570

[86] https://iq.opengenus.org/types-of-autoencoder/

[87] https://towardsdatascience.com/lstm-autoencoder-for-extreme-rare-event-classification-in-keras-ce209a224cfb

[88] Balaram Sharma, Prabhat Pokharel, Basanta Joshi " User Behavior Analytics for Anomaly Detection Using LSTM Autoencoder – Insider Threat Detection", IAIT2020, July 1–3, 2020, Bangkok, Thailand © 2020 Association for Computing Machinery