National College of Ireland

Detection of Insider Threats Based on Deep Learning Using LSTM – CNN Model

MSc Cyber Security
Internship

Tahaseen Tamanna
18193633

School of Computing
National College of Ireland

Supervisor: Mr. Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Tahaseen Tamanna
Student ID:	18193633
Programme:	MSc Cyber Security **Year:** 2019/2020
Module:	Internship
Supervisor:	Mr. Imran Khan
Submission Due Date:	17.08.2020
Project Title:	Detection of Insider Threats Based on Deep Learning using LSTM-CNN Model
Word Count:	7507 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: Tahaseen Tamanna

Date: 17.10.2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detection of Insider Threats Based On Deep Learning Using LSTM-CNN Model

Tahaseen Tamanna
18193633

Abstract

Insider threats are threats that originate within the organization and these insiders are the trusted employees in an organization. Detection of such internal attacks is challenging as the insiders have insight on sensitive information and vulnerabilities in the system. However, user actions and behaviours can help to determine the malicious activities at an early stage. Previous work mainly focused on traditional non-machine learning and machine learning techniques. Detailed research and study on deep learning techniques is achieved. One of the advantages of using deep learning approach is to have the advantage of learning features automatically. Therefore, a model is proposed with Deep Neural Network to detect insider threats based on user behaviour. Combination of LSTM and CNN (Long Short Term Memory and Convolutional Neural Network) is used to detect the anomalous behaviour of the users.

In this paper, the proposed model for detecting insider threats is to apply the combination of LSTM with CNN to the user behaviour activity data. The proposed method is applied to public dataset CMU CERT version r4.2 of size 12GB. Result of the experiments shows the proposed model can detect insider threats successfully with the ROC 0.914 and comparison with machine learning approach justifies the proposed model can successfully detect insider threats.

1 Introduction

In today's world, the organization is always under constant threat and attacks from malicious insiders. Recently insider threat has become an enormous challenge for the business. Insider threat detection is considered as the preventive approach in 1. These threats are among the most harmful and challenging attacks to deal with in practice [1]. Since insiders are the trusted users who have knowledge and authorization to access with organization's sensitive information, this makes the consequences of internal threat worse as the insiders are well aware of the loophole and deployment of systems [2]. Privileged users making these attacks look like a typical day to day activity . Many of today's security threats are not due to the result of external attacks or malicious but instead originate from the trusted insider with privileged access to the information [2]. Recently, massive investment is being made on such threats and attacks, but it may go effortless if early detection is not carried out.

Insider threats are predominant in the current world than external threats and attacks. According to a recent report from Fortinet Insider Threat 2019 [3], 68% of organization approve that insider threats are more frequently occurring than external threats and 68% feels the organization is hugely vulnerable to malicious insiders than the external attacks.
Insider threats are originated from the internal source due to the following reasons.
a. *Malicious insiders* can perform malicious activity against the organization to create harm for their benefits.

b. *Negligent* due to lack of security awareness and training to the employees, which may tend to harm the organization unintentionally. Negligence is considered as the most expensive type of employee risk.
c. *Collusive Collaborating* with the malicious external threat to gain access to the inter treasure of the organization. Though this was rare, an increase can be seen in recent times. A disgruntled employee aims to sabotage an organization by combining external factors or at the time of their resignation or even after they leave the organization.
d. *Third-Party* vendor or third-party business having crucial access to the network can also play a vital role in the compromise of the organization.
e. *Accidental* phishing attacks frequently trick employees into sharing sensitive information about the organization by posing a legitimate business or trusted partner or attachment comprising of malware.

For the successful occurrence of the attack, three elements are considered: opportunity, motive, and ability [4].
Opportunity: Threat is originated based on the opportunity level. High the level of opportunity, high the chances of performing the attack. This is based on the user's role in the organization and his activities features.
Insider's role: Each user is assigned a role linked with a user account. The role defines a different kind of access and privilege to perform certain specific tasks.
Activity Based: User's activity can be recorded in log files. These log files, which are real-time data, are the main source to detect insider's malicious activity. The main categories are- file, database, e-mail, HTTP domain, device, and network flows.
Elements of insider threats and attacks is shown in Figure 1.

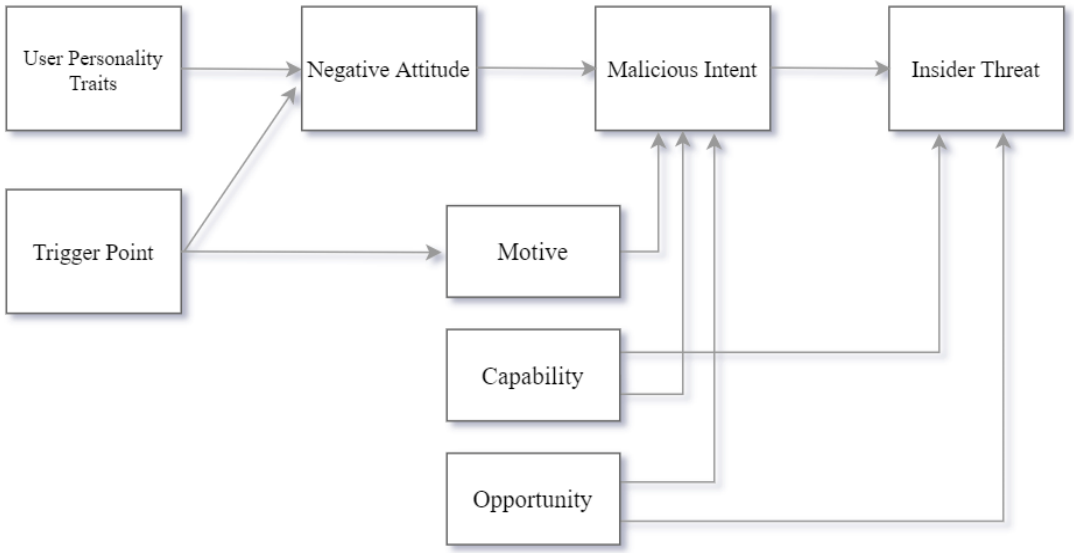

Figure 1: Overview of Insider Threats

Motive: It indicated the reason to initiate the attack. Existing research shows that motivation factors can broadly be classified into the following [5]
Emotional State: Negative state of mind such as anger, stress due to personal choice.
Personality Factors: Multiple personality traits of a user can be determined on the basis of language used in e-mail communication, web browsing, and Language Style Matching (LSM)
A tendency to malicious Activity: This can be captured by the deployment of honeypots to detect malicious behaviour. When non- privileged users interact with honeypots, which is the dummy set-up, it is considered to be an anomaly activity.

Capability: This determines the potential of the user to what extent a system or a piece of information can be compromised when the opportunity arises. This is further classified into three levels which shows how sophisticated the insider is.
Different application usage by the insider.
Multiple usages of unique application at the same per session.
CPU and RAM consumption, which is higher than generally required

Following research question is studied and addressed.

Research Question 1: How Deep Learning techniques can be applied on user behaviour activities to detect insider threats based on combination of LSTM and CNN model?

Research Question 2: How finding user anomalous behaviour can help detecting insider threats based on Deep Learning techniques?

Research Question 3: How CNN-LSTM model detect insider threats on textual based representation of user behavioural activities ?

There are vital challenges prevalent in insider threat detection is unbound patterns, since detection is time-related phenomena, dynamic environment, false alarm, highly imbalance of class, and undetected insider threats. Previous traditional ML approaches still unable to work on user behavior data with full potential due to high dimensional, complexity issues, and heterogeneity. Deep Learning concepts have attracted attention recently to detect insider threats by overcoming the challenges discussed above [6].

In this research, we aim to describe how the data from various log files based on insider's behavior and activity are processed and trained using Deep Learning Techniques to detect the malicious action leading to insider threats. The advantage of Deep Learning is utilized with the combination of LSTM and CNN to detect the malicious activity of insider threats. The neural network provides an efficient result based on vector representation [7]. The goal of the proposed model is to detect malicious behavior of users in relatively less time, which is not addressed in previous research along with minimized false alarm rate and high detection value.

The rest of the paper is structured and formatted as follows. In section 2, a brief review of Deep learning and its importance in the field of insider threat detection. The survey of all the related works and approaches are highlighted in section 3. A detailed explanation of the proposed model, including pre-processing of data, training LSTM for feature extraction, and CNN for insider detection is presented in section 4. The overall process flow and architecture of the proposed model, with all the related approaches, are shown in section 5.

2 Literature Review

The literature was searched through the keywords of insider threat detection and prediction using Google Scholar Database.
The found articles were furthermore scrutinized based on below mentioned following questions. If all the answer is yes, then the detailed analysis of insider threat detection was carried out using non-machine learning, machine learning, and deep learning techniques and approaches.

a) Is the paper/article related to cybersecurity and data science?

b) Is the paper related to current trend in detecting the insider threats?
c) Is the paper focuses on machine learning and deep learning approaches?
d) Is the paper/journal in English?

2.1 Non-machine Learning Approaches

There are varieties of research on insider threat detection and how this can be used to prevent its occurrence at an early stage. Won Park, Youngin You, and Kyungho Lee [8] in the research stated that insider threat is classified into two types of researches: technology-based and psychology-based. Salem et al. [9] probably was the first one to propose a model based on the concept of user profiling – network and host. In his research, he divided the internal user as a masquerader, traitor, and unintentional perpetrator as a technique to detect anomalous behavior of the user. Zeadally, Yu, and Lily [10] have discussed various solutions, trends, and challenges. The research paper comprised a plethora of methods from Intrusion Detection System, Exfiltration techniques, System Call, and Honeypot, in which these techniques fall under the category of network and host user profiling.

A Virtual Honeypot Framework was proposed by Niels Provos in his paper [11]. Niels Provos researched and designed the virtual honeypot framework to detect threats which are either external or internal. Honeypots mainly used to find the intruders by detecting, capturing, and analyzing external threats as a part of perimeter security Honeypots are basically dummy setup tricking the intruders to act as a live network or production server. Provos worked on a model based on honeyd, focusing mainly on the detection of external threat [11]. Main advantage of using honeypots is maintenance cost which is low and number of false positive is less compared to network intrusion detection system (NIDS) but this framework is limited upto external threat detection providing network perimeter security .

All honeypots have the same purpose and set up of no production value. Lance Spitzner, who is the founder of HoneyNet project proposed a paper where honeypots can be used as a preventive measure for insider threats rather than external threats [12]. Here, the attacks from the user is directed to the honeypot without the knowledge of the users or employees so based on the activity, it can monitor their action. As per the paper, author worked on two different strategy to deploy more realistic honeypots. The main challenges was redirection and deployment of more realistic honeynets in-order to overcome the problem, the author in proposed paper used honey token and honey nets which is combined to form adaptive behavior of users. These honey token are customizable and adaptable to the environment. As the user approaches the honeypots, it is then directed to honeynet which is used to enumerate more advance information about the user's intent and behavior.

According to the author Lance Spitzner, honeynets has been used to capture external attacks drastically and same concept has been applied in his research to record insider attacks [12]. These honeynets has the capacity to work under the dynamic environment. However, honeypots deployed is low level interaction which provides limited information about the intruder and it is longer useful if it has been exploited or detected by the users. This challenge was solved by Bertrand Sobesto [13] worked on various types of honeypots and various other types of data and log collection devices. This paper researched on visual representation of the flow of the attacks. Multiple honeypots were proposed by Farouk shah [14] and researched on open source technologies can be effectively used to detect threats. Though detection rate was good honeypot cannot be the sole technique to detect threats. Hence centralized log management tool such as logstash was implemented to collect logs from honeypots. Based on review done, it is known that non machine learning are not effective to detect insider threats in real time.

2.2 Machine Learning Approaches

The Related work here implies to research based on supervised and unsupervised machine learning techniques. Many existing research focused in the area of insider threats and attacks such as Sanzgiri and Dasgupta [15] on a recent survey explored the classification of the detection techniques based on (1) anomaly-based, (2) access-control, (3) scenario based, (4) risk analysis - workflow, (5) improvising the network defense, (6) improvising access control defense, (7) process control to prevent insiders and (8) risk analysis – psychological factors.

In the research paper, Pallabi Parveen [16] highlighted on stream mining and insider threat detection. The research also contributed on proposing a supervised machine learning model based on one-class SVM. Data gathered and collected over a longer period of time is converted to chunks. Two models are built using onc-class SVM and two-class SVM and comparison is made between the both. Insider threat data is dynamic and continuous so the pattern keeps changing over the time. Therefore, this model has a limitation of length, static data and does not adapt to changing pattern.

Another supervised machine learning technique was proposed by Sarma et al [17] based on K Nearest Neighbor (KNN). In this paper, users are classified into 4 different groups such as legitimate users, possibly legitimate users, possibly not legitimate users and not legitimate users. Model was proposed to where possibly not legitimate and not legitimate users were filtered through facial recognition [17]. KNN algorithm was considered as better in terms of performance when compared with Back-propagation Neural Network and Decision Tree [18]. However, KNN algorithm has the major drawback on efficiency as the dataset grows, sensitive to outliers and imbalance dataset.

Another common approach in detection of insider threat is framing it as an anomaly detection. Anomaly and user behavior analysis is the most common approaches [15]. Chandola, Banerjee et al. [19] proposed a research on anomaly detection. Machine learning approach can be effectively used to find anomaly activities to detect the insider threats. Ronao and Cho [20] proposed a model based on Random forest to identify the anomaly behavior related to Database Management System. The model used principle component analysis to extract features. The proposed model had the challenge of information loss, although PCA tries to cover entire variance it may lose some important data. Overall, supervised machine learning techniques has the disadvantage in classification large data and training requires high computation time.

A large amount of data is processed by an organization which is dynamic in nature. This sometimes can be unsuitable for pre-defined and in-built framework to detect insider threats. Rashid [21] proposed a unsupervised model to set normal activity as a baseline and captured user's activity sequence for a fixed period of time. Any deviation from the baseline is considered as the potential insider threat. Tiwari and Shrivastava [22] proposed a hill climbing algorithm to overcome the problem of k-means algorithm. K-means is a unsupervised algorithm, where it is necessary to know the number of clusters before the training. Hill climbing algorithm determines the local optimum which is used to find the number of clusters to be used by K NN algorithm. However, the paper fail to test the efficiency of the two algorithms. Gavai et al. [23] used network logs to detect insider threats and collected details of most used features to help isolating of data sample in the tree to understand the cause of user being tagged as malicious. A comparison was carried out between the supervised classifier approach with unsupervised approach that is isolation forest. Another comparison between supervised and unsupervised techniques was achieved by Duc C [24] in the research. Self-organizing Maps (SOM) was proposed and compared it with Hidden Markov Models, and Decision Tree to evaluate the insider threat detection. The result of the comparison shows that SOM provide better output than the rest.

2.3 Deep Learning Techniques

The challenges of insider threat detection is studied and various solutions have been proposed in terms of machine learning. The old detection approaches which depends on feature engineering are time consuming and difficult to differentiate the activity between normal and abnormal because of the characteristics of data such as problem in labelling insider threat, complexity, dealing with variety of data [6]. Deep learning is introduced for such data to provide a new standard to understand end to end models from high complex data. DL is used as an efficient technique to analyze the user behavior to detect the malicious activities.

Recently, deep learning techniques such as Convolutional neural network (CNN), Recurrent neural network (RNN), and Graph neural network (GNN), approaches have been proposed to detect insider threats. Also, various research on Deep Feedforward Neural Network is done such as deep autoencoders, Deep Belief Network (DBN), and Deep Boltzmann Machine (DBM) [6].

Deep autoencoders comprises of encoder and decoder. Work of encoder is to encode the input data to hide the representation and decoder attempts to reconstruct the input data. The main goal of deep encoders to make reconstructed data as close as original data [6].

Liu et al. [25] in his paper, an ensemble of deep autoencoders to perform the task of score calculation based on error from the output of original and reconstructed data is proposed. Feature extraction is based on the concept of frequency approach and user behavior can be analyzed from a list consisting of attributes which are extracted from various category of the data file. Individual model is designed from each autoencoder for the input features and finally, all the all the models from autoencoder is combined to build one potential model that can analyze user behavior. This proposed method resolve the nonlinear relationship in the data but the entire process requires more time. Feature extraction using frequency-based approach do not provide expected result and other main challenge is data log from different source cannot provide much efficient outcome in the proposed method.

Lin, Zhong et al. [26] worked on a method using hybrid Deep Belief Network (DBN) and Restricted Boltzmann Machine (RBM). The model is set into two phases: using DBN for feature extraction and using SVM to detect insider threats. The first phase includes learning of feature extraction using DBN, then the first layer of DBN is used to extract and train hidden features followed by multilayer RBM. Back propagation is set-up that receives the output feature vector from the RBM. The second phase includes these features as input to SVM to train the detection of insider threats. Zhang et al [27], also worked on DBN model but used unsupervised approach. Feature extraction from the behavior activity logs is in the form of a tuple including occurrence time of the activity, subject of the behavior, host responsible for the generating the behaviors, and other specific behavior observed. The process works on the basis of three items familiar to it and fourth depends on behavior types. Once the fourth behavior item is determined, normalization on the extracted feature is achieved by 1/n code discretization. Normalized featured are fed as input to the DBN, which is always built on more than one RBM hidden layers using sigma function. The proposed model faces the challenge of integrating the determined fourth behavior item with the rest of the tuple.

Advantages of using Recurrent Neural Network is sequential data modeling. The user behavior activity to detect insider threats can be modeled as sequential data. The fundamental concept is to train RNN model to predict the possibility of next activity. If the predicted outcome and the real user activity matches up to extent then it is considered normal else abnormal activity [6]. Choras and Koriz [28] proposed a model based and Deep Neural Network and Recurrent neural network. Unsupervised approach has been used to detect the malicious activities in the logs. This approach focused on the efficiency of high velocity with automated generation of result to minimize the human interference.

However, RNN approach usually faces the problem of exploding gradient. Long Short Term Memory (LSTM) is used to overcome this challenge. RNN-LSTM model was proposed in both the papers [29] [30]. Tuor et al. [29] used network logs where the features were extracted with one vector a day. This extracted features were the input to RNN which first learnt the normal behavior and then predicted the next course of action to be malicious activity by using LSTM. This method was compared with other algorithms like SVM, PCA, and Isolation Forest to evaluate its performance. Meng et al. [30] also worked on the same methodology and comparison but additionally used Kernal PCA for the analysis and detection of insider threats [6]. CNN is potentially used to work with image classification. Mouse based behavior of user is captured as an image and CNN model is used on this image file to detect insider threats [31]. Gayathri R G and et al. [32] proposed a model to detect insider threats based on image based user behavior. The feature extracted from user behavior file is presented in gray scale images and these images determines the behavior of users. Each image is fed as input to CNN layers using Mobilenet. In the paper proposed by Tara, Oriol et al. [33] combined CNN, LSTM, and DNN into one model taking all the advantages as an unified solution. Ahmed Saaudi [7] proposed model focuses on Natural Language Processing and CNN. The author used NLP for improving the performance of the model as it enables to perform classification.

Table 1: Deep Learning Approaches discussed

Model	Training	Paper
Deep Feed-forward Neural Network	Unsupervised	[25], [27]
	One-class SVM	[26]
Recurrent Neural Network	Unsupervised	[28], [29],[30]
Convolution Neural Network	Supervised	[31][32][33][34

Based on the literature survey done, it is highlighted that Deep Learning have the potential properties and advantages to proceed further with Deep Learning techniques. Following are the advantages highlighted: Deep Learning can process large dataset with an ease, feature extraction is discovered automatically required for detection, it can capture complex user activity log to determine the behavior pattern to differentiate normal and abnormal pattern. Comparing with non-machine learning and machine learning, deep learning models are more powerful in detecting insider threats.

3 Research Methodology

3.1 Proposed Model

A detailed explanation of proposed model is achieved in this section. The proposed model comprises of Deep Neural Network set up in two phases . The first phase is using Long Short Term Memory and second phase is CNN. In the paper [33], shows the combination of LSTM and CNN to detect speech recognition and recently in the paper [7] shows CNN can be used for textual classification for detecting insider threats. This has motivated to use the combination of LSTM and CNN to use insider threat. The proposed model uses LSTM to extract the features vector and these features are fed as input to the CNN to detect insider threats. The model comprises of 4 parts – i. Data pre-processing, ii. Feature Extraction, iii. Fixed size presentation, iv. Classification [34].

3.2 Pre-processing of Data

The initial phase of the research is collection and preprocessing of data. The data is collecte d from CMU CERT Insider Threat. The data consists of two main categories such as user's activity and the structure of the organization. CMU dataset consists of five log file records. These are raw files that requires cleaning and transformation.
Pre-processing consists of 4 steps: Filter, Merge, Extract and concatenate. The CMU CERT Insider Threat dataset version 4.2 is large dataset of 12GB and therefore, the process of filtering the outliers and pre-processing is challenging and time consuming.

3.3 Feature Extraction

Anomaly detection using Deep Learning considers the transformation of large heterogeneous log lines to numerical features [29]. Logfile consisted of – logon, file, device, HTTP, email. The logon.csv file used to extract user system usage on each day, length of work, which leads to user's access pattern. File.csv contains the different types of files and type of operation being performed on it by the user.
Http.csv file contains the browsing and surfing information of each user.
Similarly email.csv file contains all the incoming and outgoing emails. Device.csv contains external devices to connect and disconnect status. Crucial challenges faced during this phase: (1) data was not pre-labeled, which means no normality baseline was fixed (2) extremely large dataset 12 GB.

A large dataset requires a large memory size, which has difficulties in data processing. Therefore, 12GB of the dataset was processed chunk by chunk, which was then processed sequentially. The following explains the process of data cleaning and feature extraction. The classification is characterized by user activity per day.

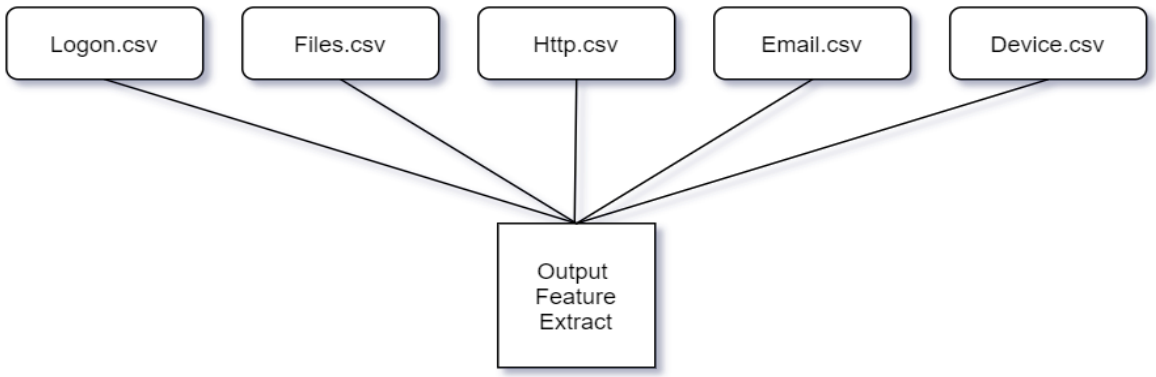

Figure 2: Feature Extraction from various log files

- HTTP: in each record of the keywords of the visited webpage were matched against the pool of selected keywords in-order to classify the user activity as
- Neutral website
- Hacktivist website
- Job Search Website

- Cloud storage
- Email: The field “TO” was used to check the recipients list belonged to the company, in order to classify as internal or external.
- Logon, Device, and File: Activities were explicitly provided. User-id, PC-id, activity, and time stamp were selected and concatenated to form the combined collection of data.

Using all the log files mentioned above, a feature vector is built by Feature Extraction. A comprehensive set of 32 actions users has performed over a fixed period of time mentioned in the proposed model. These activities are based on the user activity sequence. It is to highlight that user activities on weekends have been omitted since the activity varies on weekends and weekdays.

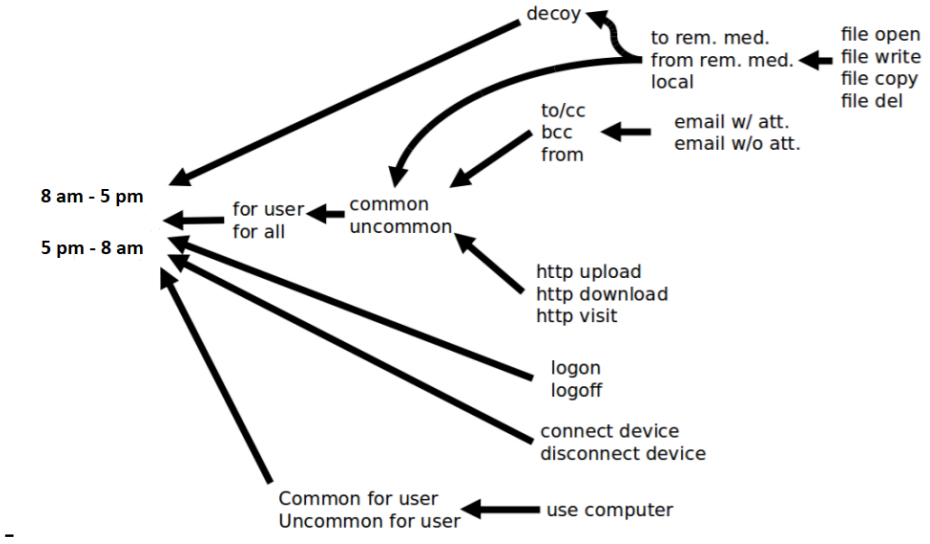

Figure 3: User Activity - Enumeration

3.4 Sampling of Data

The dataset used in the proposed model consists of more instances of non-malicious when compared to that of malicious. Therefore, it justifies that the distribution of data is highly imbalanced. Imbalanced data can be managed with various methods such as classification of algorithms. A sampling of data is mainly done to achieve a better balance. The data is sort in terms of user and data. Every user record per day was examined and

The dataset is divide into two: training and testing. 70% of the data is used for training and model selection, whereas the remaining 30% is for testing, which includes performance evaluation.

3.5 Long Short Term Memory (LSTM)

The data set is merged and aggregated as one single mega data set. Depending upon the user activity action time sequences, a feature extractor is built which has the capability to extract a feature that is associated with time series of each input action. LSTM consists of 4 layers which are 1. Input layer 2. Output layer 3. Embedding layer and 4. LSTM layer. LSTM falls under the category of RNN but the main problem in RNN is higher the distance between

two sequence higher the value. Also, on temporal sequence, difficulty level increases to train. This leads to gradient problem of exploding [7].
Therefore, LSTM model is used to overcome the above mentioned problems. LSTM memory cell is structured as follows:

- Input gate: New information is led into the system
- Memory cell: Stores the hidden information across time sequence.
- Forgot gate: Information that is no longer required is removed.
- Output gate: This contain the output that will be proceeded forward or provide activation as a new hidden state till n number of times.

LSTM solves two major problems of RNN:

- Dependency problem
- Gradient exploding problem

Detailed working of LSTM model is explained in the section 5.5 along with the implementation. Architecture of the LSTM is shown in Figure 4.

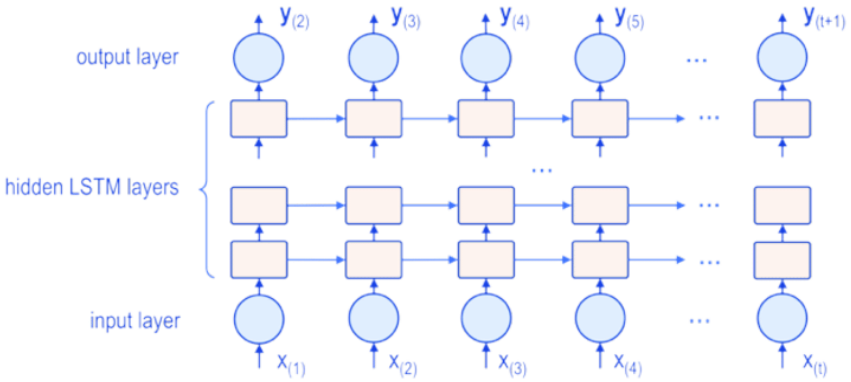

Figure 4: LSTM - Workflow

3.6 Convolution Neural Network

The final entity of the process is the training of CNN. This is also the classification stage. CNN Model consists of the following layers. (1) Input layer: This layer contains the input. The extracted feature is converted to fixed-size representation. (2) Convo layer: A piece of the input is attached to the convo layer. This performs the convolution operation. (3) Pooling: It is used to decrease the spatial volume. (4) Fully connected: It is used to classify text by providing training. Based on this, it determines the probability of user behaviour is normal or abnormal. The anomalous activity of the user leads to the detection of insider threats

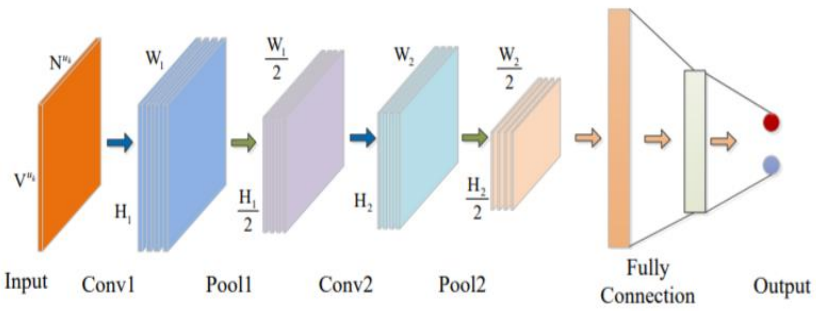

Figure 5: Layers of CNN

3.7 Comparison of Proposed model with SVM

Selecting the right technique to detect insider threats is a crucial task. Once the technique is experimented, it is always necessary to compare it with other algorithms to under the performance of proposed model. In this study, the proposed model's result is compared with SVM to analyze its performance and determine if it is an efficient model. Based on the analysis conducted by Iffat and Ali [2], which shows the number of machine learning algorithms used so in the detection of insider threats. The survey presents that graph learning algorithms are the most commonly used algorithm followed by Support vector Machines.

Table 2: Classification of Algorithm based on Number of Times used for Insider Detection

Different Learning Algorithms	Number of Algorithms Used
Regression	1
Game	1
Gaussian Mixture Model	1
K- Nearest Neighbor	2
Fuzzy Interference Systems	2
Support Vector Machines	4

4 Design Specification and Solution

4.1 Proposed Model Process Flow

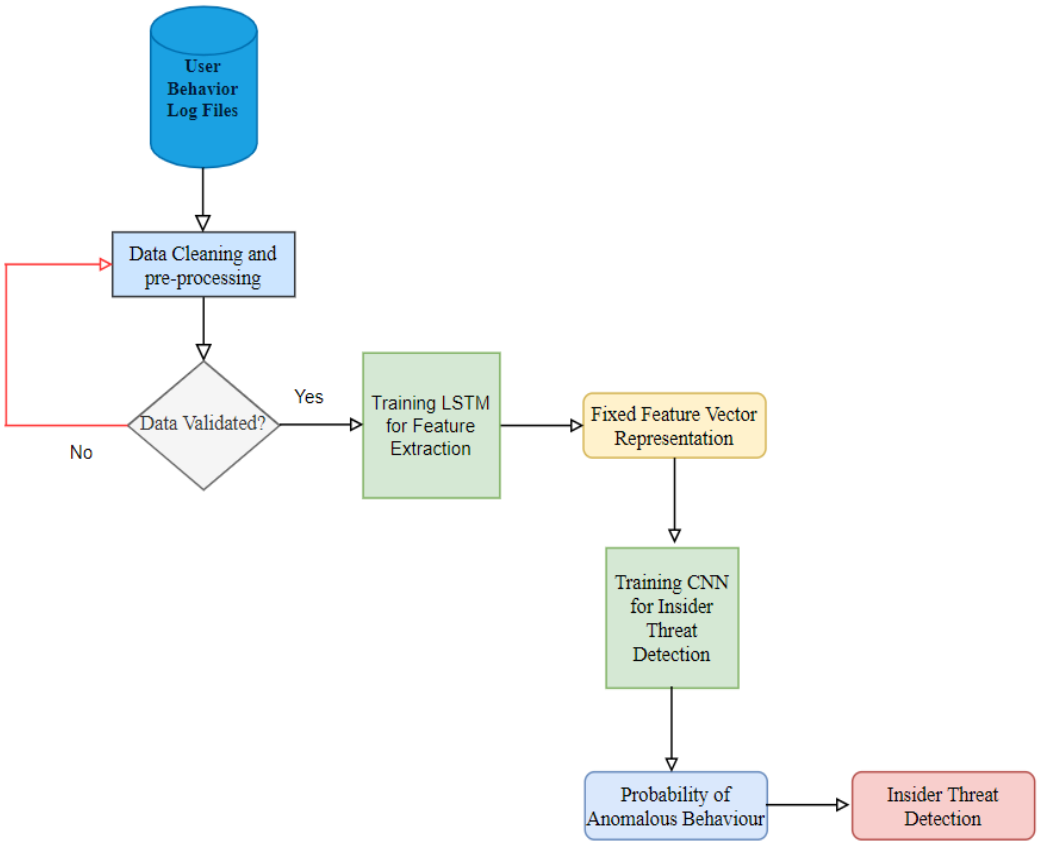

Figure 6: Process Flow

4.2 Insider Threat Based on Behaviour Analysis

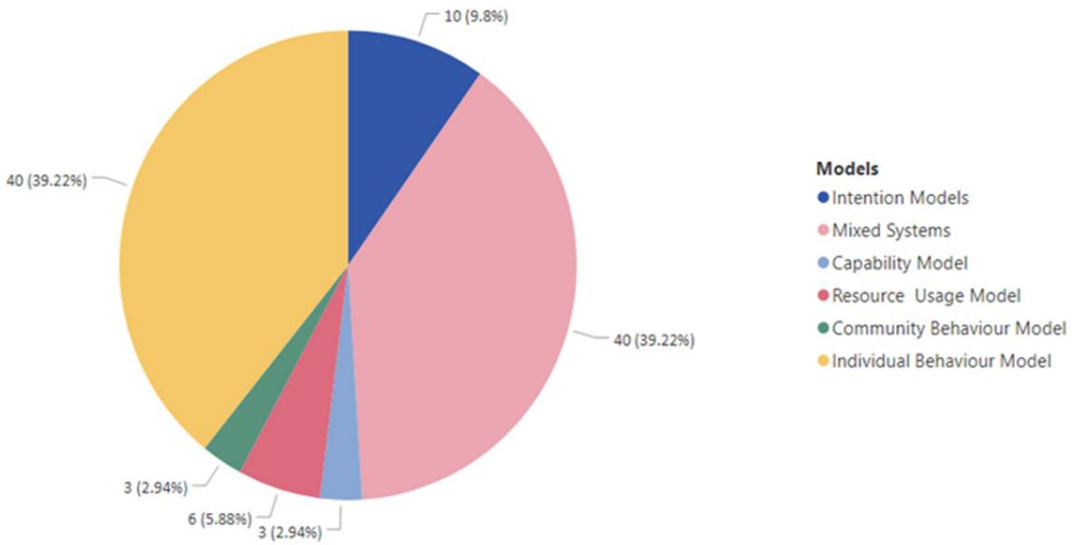

Figure 7: Classification of Insider Threat Models based on Feature

Depending upon the features used and extracted, the model is built. Insider threat detection contains six model categories.

Based on the survey conducted in the paper [2], it shows the majority of existing internal threat detection methodology, and models use either the Individual Behavior Model or Mixed System Model, accounting for approximately 80%. Hence, Individual Behavior Model is used in this proposed research to extract features and develop the model.

Table 3: Insider Threat Model

Intention Model	Capability Model	Resource Usage Model	Individual Behavior Model	Community Behavior Model	Mixed System Model
Based on user psychological behavior	Based on usage of resource by insiders	Information of users on specific resources.	User day to day activities and course of actions		Combination of all the models together

4.3 TensorFlow Keras Framework

Implementation of the proposed method requires a deep learning framework. Various deep learning frameworks are available. Tensorflow by Google is known for its graph, scalability, and allowing the monitor to the status of training models. Additionally, high-level API can be placed above the TensorFlow to make the learning more potential. In this paper, Keras is used, which is most commonly preferred. Multiple layers are connected in the form of sequence to potentially process Tensor. Following is the list of layers available in Keras: Dense layer, Dropout Layer, Embedding layer and Recurrent layer, Pooling layer, Convolutional layer.

4.4 Architecture of the proposed model

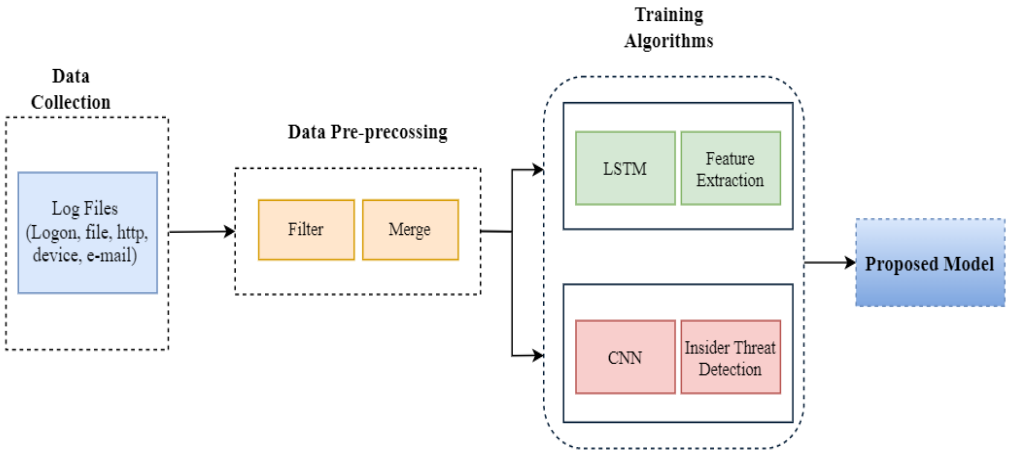

Figure 8: Architecture LSTM-CNN

5 Implementation

This section works on experimental validation and evaluation of the proposed model, implemented in Tensorflow and Keras using Google Colab. It is an open-source platform to perform deep learning implementation. A detailed explanation of the implementation is given in the following section.

5.1 Dataset

In the research, the proposed model is applied and experimented on CMU CERT insider threat dataset version 4.2 to simulate the model. CMU CERT Insider Threat Dataset is explicitly selected as this is the standard dataset to perform detection of insider threats and anomalous behavior. Comparatively, version R4.2 consists of more instances compared to other versions with the size of 12GB. This is a collection of the synthetic dataset [35].
This dataset comprises of activity logs over a period of 17 months for 1000 users. The proposed system utilized five events/activities – i) logon/logoff activity ii) HTTP website browsing activity iii) Email incoming and outgoing communication iv) File operation activity v) Device information.

Table 4: Overview of CMU Dataset Details – Statistics

Users	1000
Period	17 months
Overall Instance	32,770,227
Malicious	7323
Non-malicious	32,762904
Insiders	70

The overall instance in the dataset however, changed after pre-processing and sampling to 32,120,072.

Parsing of each log line, and an instance is done to acquire the user activity details. The number of actions over different types of activities is 32 and is based on action sequences.

5.2 Training of LSTM for Feature Extraction

Information in the LSTM can be carried across sequence; therefore, information can be saved for the future without the problem of vanishing old signals. The structure is explained below

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad \text{..... (1)}$$

$$f_t = \sigma(W_{xf}x_t + W_f h_{t-1} + b_f) \quad \text{.....(2)}$$

$$c_t = f_t * c_{t-1} + i_t * tanh(W_c x_t + W_{hc}h_{t-1} + b_c) \quad \text{.....(3)}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad \text{.....(4)}$$

$$h_t = o_t * tanh(c_t) \quad \text{.....(5)}$$

Equation (1) it represent the input gate. Equation (2), ft represents forget gate. In (3) ct-1 is the output of the previous cell. In (4) and (5), the output gate and final state (ot and ft) are shown respectively.

In our implementation, each user activity with the length of the action sequence represented at the hidden layer. Our study used a one-hot encoding to perform the embedding task. User ID and actions are associated, and a dictionary is created. For example, user's sending the external email is labeled as 1, and internal email communication is 2. These actions are transformed into vectors with 1 and 0, respectively. This process contains three steps:

(1) User input series as a vector.

(2) Converting the actions to a one-hot vector in the embedding layer.

(3) Input to LSTM

(4) Sparse Categorical Cross-Entropy Loss used to compare the original answers with the output.

While training for LSTM, overfitting problems can arise for which Dropout is applied. The number of the unit were assigned to 40 and training was performed using 10 Epochs, where each epoch considered randomized order and had the batch size of 20.

Table 5: Parameters – LSTM

Training Model	Hidden Layer	Batch Size	Epoch	Drop-out
LSTM 1	40	20	10	0.2

5.3 Fixed Size Feature Vector Representation using Embedding

CNN model accepts fixed-size vector representation as input, and this is achieve by using embedding methodology. Before feeding input to the CNN model, it is important to vectorize the text. Embedding builds a fixed size matrix and represents the categorical variable to the vector of continuous numbers. This method solves the different session problem.

A maximum and minimum length are assigned for the action sequences. Now, there are three cases that have to be considered. (1) When the sequences length is lesser than the minimum length, then those sequences are omitted. (2) For the sequence whose length is greater than the maximum, only the first value is considered. (3) for the sequences length between the minimum and maximum length, each input sequence is zero-padded to the count of maximum length. Therefore, achieving the same equal length.

In the figure, Nuk is considered at the first value of the maximum length [34]. For generating embedding in this study, 150 activities per day were assigned, and the hidden state of the third LSTM layer for each input is concatenated to get a vectorized text size of 150*40, which is the representation of 2D matrix.

5.4 Training CNN for Insider Threat Detection

Classification is the final stage of the process. This phase determines normal behaviour with abnormal by classifying fixed-size vector features. The 2D represented matrix is assigned as input to the conv1 layer. The output layer consists of two dimensions: normal or anomaly. CNN is first trained with fixed-size feature along with the explanation of user behaviour and its normal/anomaly activities.

In this study, 32 and 64 filters are the numbers applied to filter. Followed by max-pooling, which is used to reduce the input size into half, is assigned to 2 and activation as tanh. The input to CNN is already vectorized to equal length along with the allotted parameter, works on the concatenation of all feature map vector to form the matrix. Training is based on ADAM optimizer to optimize the network. After which, trained CNN can be used to determine the probability of anomalous user activity.

Table 6: Parameters of CNN

Training Model	Conv1	Conv2	Activation	Batch size	Epoch
CNN	32	64	Tanh	512	70

6 Evaluation

6.1 Accuracy

The overall performance of the proposed model evaluated using accuracy. It is one of the evaluation metrics for the classification model.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

In our proposed model, the accuracy for training and testing is 0.94 (94%) and 0.95 (95%), respectively. Since the accuracy values of both train set and test set is almost similar, the proposed model does not overfit the training data.

Considering the fact that large dataset used for the proposed model is highly imbalance, hence Receiver Operating Characteristic Curves (ROC) and Area Under Curve (AUC) is used to measure the evaluation of the proposed model.

6.2 Receiver Operating Characteristic Curves (ROC)

This graph shows and evaluates the performance of the proposed classification model. This curve works under two parameters: True positive rate and False positive rate.

True Positive Rate (TRP) is defined as:

$$TPR = \frac{TP}{TP + FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2)$$

False Positive Rate (FPR) is defined as:

$$FPR = \frac{FP}{FP + TN} \quad \ldots\ldots...\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

Interpretation:

The relation between TPR and FPR is represented below using ROC. The goal here is to find a point on the curve where it has the maximum area. At this maximum point, the model can differentiate between binary classes with minimum overlap. In order to minimize the overlap, the ROC area value should be high. The graph below plots the sensitivity and specificity

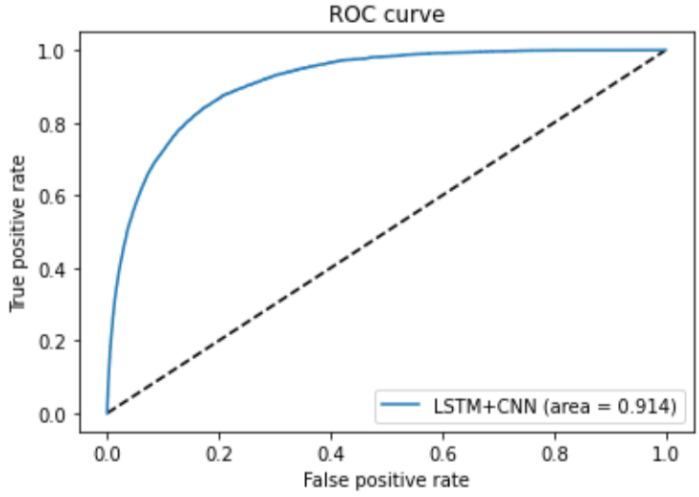

Figure 9: ROC Curve Represention

6.3 Area Under ROC (AUC)

It is the 2D area value from (0,0) to (1,1) presented under the ROC Curve. Practically, the overall accuracy and performance are collected from the classifier to provide AUC.
Here, the area value is 0.914, which is close to 1 indicates that data classes are separated properly, and the proposed model is efficient.

$$AUC = 0.914$$

6.4 Experiment / Case Study 1

Here, in this section, the result of other algorithms, such as SVM is presented. The proposed model further experimented with other machine learning algorithms to perform the proposed model performance analysis. The experimental result shows the value = 0.50, which justifies that the SVM model performance is poor. Based on the literature study, the following are observed, and comparison has been made between the proposed approach and already existing approaches.

Modified Isolation [23] with an accuracy of 82% and Deep Auto Encoder [25] with an accuracy of 90.25 shows that the proposed model is efficient in detecting insider threats.

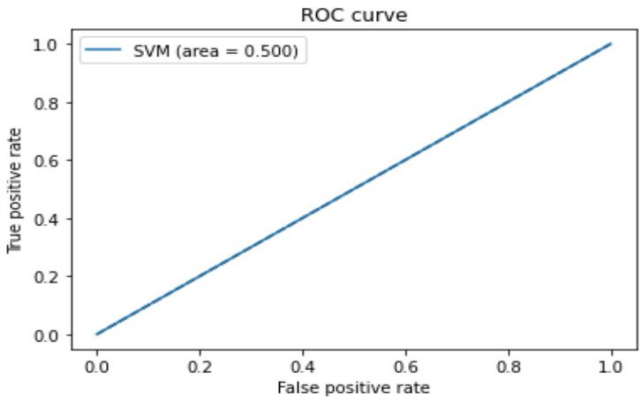

Figure 10: ROC for SVM

6.5 Discussion

Deep neural network for insider threat detection is proposed. This is achieved by fixing a baseline and train the model to determine normal and anomalous behavior leading to insider threat detection. This is based on the approach of LSTM and CNN. The proposed model has been improvised and efficient when compared to the other existing work. This is justified by the results obtained from the implementation and experiment values. The existing model based on Modified Isolation [23] and Deep Auto Encoder [25] have an accuracy of 82% and 90.25%, which shows that the proposed model achieves maximum productivity. The comparison with SVM also highlights that LSTM-CNN is an efficient approach for detection.

Based on the related works, the first phase of the research to work with the deployment of medium level interacting honeypots to capture the malicious activity of users. It is followed by a centralized Log Management tool like Logstash (ELK stack) to analyze the raw logs from honeypots. These raw logs are converted to a human-readable format dataset. The real-time generated logs to be trained by neural networks to detect and predict insider threats. However,

it was found that due to limited time, a sufficient amount of log files required to properly train the model could not be generated. Since deep learning requires large data, this approach had a limitation.

In our study, CMU R4.2 synthetic dataset is used, which is large and highly imbalanced, consisting of 3.2 million records. Therefore, the proposed model can still be improved in the cleaning, pre-processing, and sampling of data to achieve a better balance. SMOTE algorithm (Synthetic Minority Over-sampling Technique) can be used for this purpose.

7 Conclusion and Future Work

Detailed study and evaluation of the deep learning technique to detect insider threat is achieved successfully. Deep Neural Network models were successfully trained using LSTM and CNN to determine the anomalous activity to detect insider threats from the features extracted out of the CMU CERT Insider Threat data logs. The approach have been applied on a large dataset to train and test the performance of the model. Experimental result shows CNN-LSMT is one of the efficient methodologies to perform the detection of insider threats.
The following is the list of key findings

- Deep neural network can be successfully used in the field of cyber security.
- Apart from image, face, and speech, CNN can effectively be used for 1D data to detect insider threats.

Execution time is relatively high in the proposed model. As a part of future enhancements to the proposed model, future work can include detection of insider threats based on gating mechanism (GRU). For future work, a model based on 2D representation of data consisting facial and mouse behaviour pattern to detect anomaly activity of users.

8 References

[1] F. T. J. G. Ivan Homoliak, "Insight Into Insiders and IT: A Survey of Insider Threat Taxonomies, Analysis, Modeling, and Countermeasures," *ACM Computing Surveys,* April 2019.

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

[3] Fortinet, "2019 Insider Threat Report".

[4] A. M. N. V. Miltiadis Kandias, "An Insider Threat Prediction Model," *Information Security & Critical Infrastructure Protection Research Group,* 2010.

[5] B. O. G. M. C. S. Legg PA, "Automated insider threat detection system using user and role-based profile assessment," *IEEE System Journal,* 2015.

[6] S. a. Xintao, "Deep Learning for Insider Threat Detection: Review, Challenges and Opportunities," May 2020.

[7] A. H. a. Z. Al-badi, "Insider Threats Detection using CNN-LSTM Model," April 2019.

[8] Y. Y. a. K. L. Won Park, "Detecting Potential Insider Threat: Analyzing Insiders' Sentiment Exposed in Social Media," July 2018.

[9] S. H. a. S. J. S. M. B. Salem, "A survey of insider attack detection research," *Springer,* 2008.

[10] Y. L. Sherali, "Detecting Insider Threats: Solutions and Trends," *Information Security Journal: A Global Perspective,* 2012.

[11] Niels Provos, "A Virtual Honeypot Framework," in *13th USENIX Security Symposium,*, August 2004.

[12] H. T. I. Lance Spitzner, "Honeypots: Catching the Insider Threat," *IEEE,* December 2003.

[13] M. M. Bertrand Sobesto, "DarkNOC: Dashboard for Honeypot Management," in *Proceedings of the 25th international conference on Large Installation System Administration*, December 2011.

[14] F. Samu, "Design and Implementation of a Real-Time Honeypot System for the Detection and Prevention of Systems Attacks," 2016.

[15] D. D. a. D. Dasgupta, "Classification of Insider Threat Detection Techniques," *Proceedings of the 11th Annual Cyber and Information Security Research Conference,* 2016.

[16] Z. R. W. B. T. Pallabi Parveen, "Supervised Learning for Insider Threat Detection Using Stream Mining," *IEEE 23rd International Conference on Tools with Artificial Intelligence,* November 2011.

[17] Y. S. M. A. L. U. M. S. Sarma, "Insider Threat Detection with Face Recognition and KNN User Classification," *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM),* 2017.

[18] a. W. M. Wenjuan Li, "Enhancing Collaborative Intrusion Detection Networks against Inside Attacks using Supervised Intrusion Sensitivity-Based Trust Management Model," *Journal of Network and Computer Applications 77:135-145,* January 2017.

[19] A. B. Varun Chandola, "Anomaly Detection: A Survey," *ACM Computing Surveys 41(3),* July 2009.

[20] S.-B. C. Charissa Ann, "Anomalous Query Access Detection in RBAC-Administered Databases with Random Forest and PCA," *Information Sciences 369,* 2016.

[21] I. A. Tabish Rashid, "A New Take on Detecting Insider Threats: Exploring the Use of Hidden Markov Models," in *MIST '16: Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats*, October 2016.

[22] M. S. Susheel Kumar Tiwari, "Implementation of Improved K-Mean Algorithm for Intrusion Detection System to Improve the Detection Rate," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* 2018.

[23] Gaurang Gava, Kumar Sricharan,Dave Gunning, "Supervised and Unsupervised methods to detect Insider Threat from Enterprise Social and Online Activity Data," *Palo Alto Research Center,* 2015.

[24] D. C. Le and A. N. Zincir-Heywood, "Evaluating Insider Threat Detection Workflow Using Supervised and Unsupervised Learning," *IEEE Security and Privacy Workshops,* 2018.

[25] L. Liu, O. D. Vel and C. Chen, "Anomaly-Based Insider Threat Detection Using Deep Autoencoders," *IEEE International Conference on Data Mining Workshops (ICDMW),* 2018.

[26] L. Lin and S. Zhong, "Insider Threat Detection Based on Deep Belief Network Feature Representation," *IEEE International Conference on Green Informatics (ICGI),* 2017.

[27] Y. C. JIANGE ZHANG, "Insider threat detection of adaptive optimization DBN for behavior logs," *Turkish Journal of Electrical Engineering and Computer Sciences ,* 2018.

[28] O. Oladimeji, "Review on Insider Threat Detection Techniques," *Journal of Physics: Conference Series,* 2019.

[29] S. K. Aaron Tuor, "Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams," *Proceedings of AI for Cyber Security Workshop at AAAI 2017,* October 2017.

[30] F. L. Y. F. a. Z. T. F. Meng, "Deep Learning Based Attribute Classification Insider Threat Detection for Data Security," *IEEE Third International Conference on Data Science in Cyberspace (DSC),* 2018.

[31] W. N. Teng Hu, "An Insider Threat Detection Approach Based on Mouse Dynamics and Deep Learning," *Security and Communication Networks,* Feb 2019.

[32] A. S. Y. X. Gayathri R G, "Image-Based Feature Representation for Insider Threat Classification," November 2019.

[33] T. N. Sainath and O. Vinyals, "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2015.

[34] Y. C. Y. S. Fangfang Yuan, "Insider Threat Detection with Deep Neural Network," *Springer,* June 2018.

[35] C. M. University, Software Engineering Institute, November 2016. [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099.