Journal of Science and Technology ISSN: 2362-1443
2023, Vol. 3, No. 1, 57-65

# Insider Threat Detection using LSTM

Durga Bhandari
Faculty of Science and Technology, Pokhara University
durga.bhandari68@gmail.com

Kumar Pudashine
Faculty of Science and Technology, Pokhara University
kumar.pudashine@ieee.org

***Article History:***
Received: 9 July 2023
Revised: 17 October 2023
Accepted: 23 December 2023

***Keywords—****Internal Threat, deep learning, GAN, LSTM, DOS, DDOS*

***Abstract*—Security threats have been the major challenge for any organization. This has even been more threatening since in present days most of the organizational data are in digital form and digital data are easy to access and alter if not properly secured. While most of the threats considered are external threats like Viruses, Worms, DOS, DDOS, hacking etc. Internal threats also cannot be ignored. Many frauds, especially for organizations that perform financial transactions, are done by misusing the internal access to the data. Internal threats happen from the users who have some privileged access to the data. Finding such a threat is not only difficult but also more challenging than that from the external source. Most organizations don't give internal threats that much consideration but lately many works have been done in the field of internal threat detection.**

## I. INTRODUCTION

In any organization, nowadays, it is essential and mandatory to have data and information in the digital form. Having data in digital form has many advantages but it comes with a common threat of data security. The source of such threats are mostly outsiders however internal threats are also a thing that cannot be ignored and left unattained. Moreover, internal threats are even more complex and difficult to identify and save systems from since internal threats are from the people from the trusted ones and the system cannot be completely isolated from them and they can evade firewalls, Intrusion Detection Systems, and other security mechanisms aimed at protecting the information infrastructure from outside attacks. In various survey it was found that in 2018, 53% of the companies suffered from Internal attacks and this number was surveyed to rose to 60% in 2020 [10,11].

Data and Information are the assets to any organization in this twenty first century. Insiders are the major contributor to the data and information creation and consumption in organization. Such users may also be the source of security threat to any organization. Moreover, the damage done by insiders is even more severe and may be hard to prevent [11]. These threats arise from individuals within an organization who misuse their access privileges, intentionally or unintentionally. Detecting and mitigating insider threats is crucial for safeguarding sensitive data, intellectual property, and organizational integrity.

The major challenges for the Insider threat detection is that the Insider threats are multifaceted, involving behavioral, technical, and organizational aspects. Understanding the nuances of these threats requires interdisciplinary research. Also the insider threats are variable in nature, insiders can be employees, contractors, or partners, making their behavior diverse and challenging to predict. Unlike external attacks, insider threats often occur gradually and subtly, making them harder to detect. Damage for insider threats can have greater impact. Insider incidents can lead to financial losses, reputational damage, and legal consequences.

The model proposed in this work will help the organization to find the insider threat with the help of logs from various sources and help to prevent such threat. This research-based work has tried to contribute to the field of data security from insiders by proposing a more reliable, trustworthy and efficient model using deep learning techniques.

### *A. RNN*

RNNs are designed to handle sequential data, such as time series or natural language. They allow information to flow from one step to the next, making them suitable for tasks where context matters.Unlike traditional feedforward neural networks, RNNs have loops that allow information to persist across different time steps. They can model dependencies over time, making them suitable for applications like language modeling, speech recognition, and stock market prediction. The core idea behind RNNs is the hidden state, which captures information from previous time steps. At each time step, takes an input (e.g., a word in a sentence). Updates its hidden state based on the input and the previous hidden state. Produces an output (e.g. predicting the next word). However, RNNs struggle with long-term dependencies due to gradients diminishing during back propagation.

### *B. LSTM*

The vanishing gradient problem made it difficult for ordinary RNNs to learn dependencies across lengthy sequences. To solve this issue, LSTMs were developed [5]. Compared to simple RNNs, LSTMs are more intricate

constructed, with memory cells, input, forget, and output gates included. Long-range dependencies in sequential data can be captured by LSTMs, which makes them useful for tasks like time series prediction and language modeling. They prevent the neural network output from either decaying or exploding as it cycles through feedback Loops. LSTMs are frequently employed in applications like as natural language processing (NLP), when comprehending a word's context necessitates taking the sentence as a whole into account. Based on past data, LSTMs are excellent at forecasting future values in a time series [6].

## II. LITERATURE REVIEW

An insider threat is a security risk that comes from within an organization. It refers to the potential for an employee, contractor, vendors, or other insider to compromise the security of their organization intentionally or unintentionally. Insider threats can take many forms, including theft of sensitive data, sabotage of systems, and the introduction of malware or other malicious code. Insider threats can broadly be categorized as intentional insider threat and unintentional or accidental insider threat. Malicious insiders are individuals who intentionally cause harm to their organization. They may do this for a variety of reasons, including financial gain, personal vendetta, or ideological beliefs. Accidental insiders are individuals who cause harm to their organization unintentionally, often through carelessness or a lack of awareness of security protocols. Compromised insiders are individuals who have had their credentials or access to sensitive information compromised by an external attacker, who then uses those credentials to gain unauthorized access to the organization's systems or data.

Various related works that have been done in the field of threat detection using deep learning algorithms have been discussed and analyzed in this section, findings and methods used in the past will be used as a steppingstone to address the limitations seen in the previous methods and techniques of the learning algorithms.

The intrusion detection system can be broadly categorized as a signature-based intrusion detection system and behavior-based intrusion detection system based on working approach. In the case of a signature-based system, a signature-based intrusion detection uses predefined patterns or signatures to identify malicious activity. Signature based IDS systems are commonly used to detect malware infections, network attacks, and other types of malicious activity. They are effective at detecting known threats but may not be as effective at detecting novel or zero-day attacks, as these types of threats do not have a predefined signature [1].

A behavior-based intrusion detection system uses machine learning algorithms to analyze patterns of system and network activity in order to identify potentially malicious behavior. Behavior-based IDS systems use artificial intelligence and machine learning techniques to learn what normal behavior looks like for a given system or network, and then use this knowledge to identify deviations from the norm that may indicate a security threat. To be effective, a behavior-based IDS should be trained on a large and diverse dataset of normal system and network activity and should be regularly updated with new data to ensure that it remains accurate and effective at detecting threats.

A wide range of algorithms, including deep neural networks [18], multi-fuzzy classifiers [37], the hidden Markov method [41], one-class support vector machines [40], deep belief networks [18], linear regression [26], clustering algorithms [24], and light gradient boosting machine [36], have been used by researchers to address the insider threat detection problem. Below, we list a few of the more noteworthy studies.

According to a study by Noever [2], which examined several families of machine learning algorithms, random forest appears to provide the best results when compared to other ML models. The CERT insider threat dataset was used for the experiments, and the risk factors were taken out of the data to create a feature vector. They included sentiment analysis variables from file-access information and the content of emails and websites. They ordered these characteristics generally according to their significance.

iForest is another intriguing machine learning technique that has drawn interest [16, 17]. iForest was utilized by Gavai et al. [17] as an unsupervised anomaly detection technique; they used features taken from social data, such as online activity and email communication patterns, to identify statistically abnormal behavior. They exploit the fact that employees who plan to leave the company are more likely to initiate insider threats by using this tactic.

The authors obtained a ROC score of 0.77 for insider threat detection by using iForest to predict when employees would leave the company as a proxy for determining the likelihood of insider threats. On an online framework, Karev et al. [16] also employed iForest for insider threat detection. An all-purpose algorithm was employed to determine the best.

It has been demonstrated that applying a predictive model with an ML algorithm to an unbalanced dataset results in significant bias and inaccuracy. The dearth of empirical data and the problem of data asymmetry indicate that insider threat analysis is still a relatively unexplored field of study. The pre-processing step of balancing the dataset has been the subject of several research studies [14, 29, 34, 39]. The spread subsample technique did not significantly improve performance when used to balance datasets, according to Sheykhkanloo and Hall's [30] results. On the other hand, the approach greatly shortened the time required to construct and validate the model. Furthermore, their tests demonstrated that for imbalanced datasets, all supervised machine learning algorithms perform better than Naïve Bayes.

For insider threat detection, Orizio et al. [22] used a constraint learning algorithm. By building an optimized constraint network that emulates typical behavior, the algorithm finds threats when the cost rises above a predetermined level. Unlike most other ML algorithms, this approach has the advantage of giving an explanation for the decision-making process. To improve the outcomes, they advise employing deep learning models in the feature extraction process in addition to hand-picking features.

Gayathri et al. [35] took a deep learning approach to the problem of insider threat detection; their method performs multi-class classification by combining a generative model with supervised learning. To improve the minority data samples, they employed Generative Adversarial Networks (GAN) for data resampling on the CERT insider threat dataset. Three distinct resampling methods were applied to four distinct classification methods in order to select GAN; the GAN method was nominated because of its encouraging outcomes in comparison to the other resampling methods.

### A. *Deep Learning*

Deep learning algorithms are particularly well-suited for tasks such as image and speech recognition, natural language processing, and predictive modeling. They are also used in a variety of other applications, including self-driving cars, medical diagnosis, and financial forecasting. Deep learning algorithms are trained using large amounts of labeled data and powerful computational resources. The training process involves presenting the algorithm with examples of the task it is being trained to perform and adjusting the network's internal parameters called weights based on the performance of the algorithm. The goal of this process is to optimize the network's ability to recognize patterns in the data and make accurate predictions or decisions. Deep learning algorithms have achieved impressive results in a number of fields, but they can be computationally intensive and require large amounts of data to be effective. They also have the potential to be biased if the training data is not representative of the overall population.

Due to their high dimensionality, complexity, heterogeneity, and sparsity, traditional shallow machine learning models are unable to fully utilize user behavior data, despite the fact that existing approaches have shown excellent performance on insider threat detection [16]. Conversely, deep learning has the potential to be a useful instrument for analyzing user behavior within an organization to identify hostile insiders. Based on the deep structure of the data, deep learning is a representation learning algorithm that can extract multiple levels of hidden representations [16] from complex data.

Recently, deep feedforward neural networks, convolutional neural networks, and recurrent neural networks have all been proposed as techniques for identifying insider threats. Some of the most recent deep-learning techniques for identifying insider threats are presented in this section.

Deep learning Algorithms such as CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network) have always achieved good results. CNN model is used to detect by capturing spatial patterns in traffic data and subsequently translating them into cyberattacks converting them to grayscale images. Similarly, RNN models are used for classification Cyber-attacks by extracting temporal data patterns of system logs [3]. Related Work on detecting insider threats in deep learning involves using a two-dimensional CNN model. Additionally, there is work done using RNNs to identify malicious activity by treating traffic data as time-series data, among them, packet-based intrusion detection has used the embedded approach of novel words and Long Short-Term Memory (LSTM) where embedding of the words will extract the semantic meaning of the traffic packets, LSTM will capture sequence information present in the packet for the attack detection process.

A recurrent neural network (RNN) is a type of neural network that is particularly well-suited to processing sequential data, such as time series data or natural language. RNNs have a memory that allows them to take into account past events when processing new data, making them useful for tasks such as language translation and speech recognition.

RNNs operate by processing input data through a series of hidden units, or neurons that are connected in a chain-like structure. Each neuron in an RNN receives input from the previous neuron in the chain and produces an output, which is then passed to the next neuron in the chain. This allows the network to maintain a kind of memory of past events, which is useful for tasks that require context or a sense of temporal dependencies.

There are several different types of RNNs, including long short-term memory (LSTM) networks and gated recurrent unit (GRU) networks, which have been developed to address some of the challenges of training and optimizing traditional RNNs.

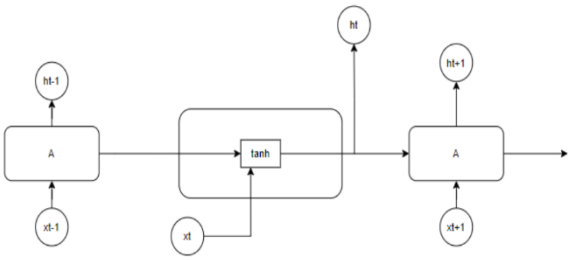


Fig. 1. Module in the simple RNN module

## III. METHODOLOGY

The proposed method is completed with the adoption of the quantitative analysis approach where associated data represents the insider activities in the simulated work environment. Both the case of normal as well as abnormal activities are used during the work, and later the developed model is able to detect the presence of threat in the insider's activity.

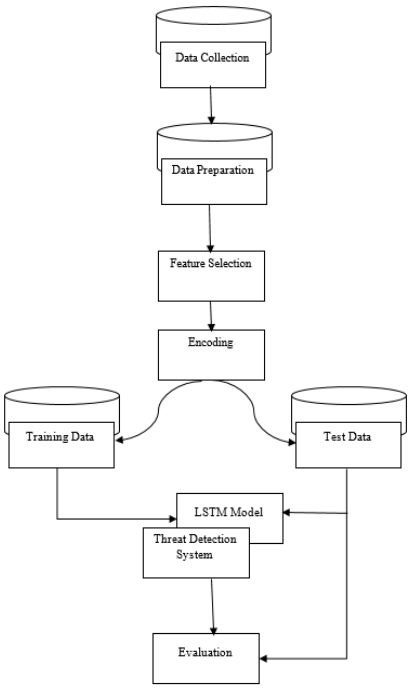


Fig. 2. Fig 2: System Block Diagram

The data used are the CERT r4.2 data. Data will be preprocessed, labelled, and feature extraction will be done. Work will be followed with encoding to convert categorical data into numerical form to feed into the model. Encoded data is separated into training and testing data. To predict the data resembling either normal or malicious data, LSTM based

classifiers will be used. Finally, the effectiveness of the work will be validated using the test data.

### *A. Data Collection*

Data collection is an essential stage in any research, the source of data should be reliable. The data can be obtained from the Primary or Secondary approach. Primary approach also known as the firsthand data collection where data is collected with the direct setup of experiments or with the help of survey or interview. Even though the data collected from the primary approach is more authentic and reliable, it takes longer time to collect such data. Accuracy obtained in the decision made using this type of data will be very reliable, thus most of the works that deal with the critical factor should be using the primary approach of the data collection. Secondary approach uses the data that are previously published and maintained and are meant for mostly research purposes. Thus, data obtained from such methods may result in conclusions that are not highly precise and accurate and hence best fit for the work that have short completion time.

| Files | Operation types |
|---|---|
| Logon.csv | Logon_Weekday (When employee logs on to the computer on a weekday at the normal work hour)<br>Logon_Afterhour_Weekday (When employee logs on the computer on a weekday but after normal work hours)<br>Logon_Weekend (When employees log on at weekends)<br>Logoff (When employee log off the Computer) |
| Email.csv | Send Internal Email (When employee sends an internal email)<br>Send External Email (When employee sends an email to external domain)<br>View Internal Email (When employee views an internal email)<br>View External Email (When employee views an email from external domain) |
| Http.csv | WWW visit (When employee visits a website)<br>WWW download (When employee downloads file from a website)<br>WWW upload (When employee uploads to a website) |
| Device.csv | Weekday device connect (When employee will connect to a device on a weekday at work hours)<br>After hour weekday device connect (When employee connects on a device on a weekday after hours)<br>Weekend device connect (When employee connects a device at weekends)<br>Disconnect device (When Employee disconnects a device) |
| File.csv | Open doc/jpg/txt/zip file (when employee opens a doc/jpg/txt/zip file)<br>Copy doc/jpg/txt/zip file (When employee copies a doc/jpg/txt/zip file)<br>Write doc/jpg/txt/zip file (When employee writes a doc/jpg/txt/zip file)<br>Delete doc/jpg/txt/zip file (When employee deletes doc/jpg/txt/zip file) |

Fig. 3. *Files in Dataset [7]*

For this work, CERT insider threat dataset r4.2 is used. CERT insider threat dataset r4.2 contains different log files in .csv format. The log files contain different data events that have been created by the users over the period. By the evaluation of user activity and the data events log, it will be possible to detect the threats that exist in any organization.

Dataset provided by CERT (Computer emergency response team) division of the software engineering institute at the Cernegie Mellon University (CMU) is used that contains 1000 case studies of the real life insider threat containing traitor instances as well as the Masquerade activities. The CERT data set contains 5 log files. Information about those log files is shown in Table 1.

Cert 4.2 dataset contains 5 different events in five different csv files. These files contains the log of 1000 employees in an organization over a period of 17 months. This dataset has 32770222 events from 1000 users with intentionally injected 7323 malicious instances. This dataset contains three primary scenarios as follows:

1. Someone who has never used a removable drive or worked after hours begins to log in, use it to upload data to wikileaks.org, and then quickly leaves the company.
2. A user starts contacting possible employers and looking for career opportunities on job search websites. They use a thumb drive to take data before leaving the office (at a rate noticeably higher than their prior actions)
3. A disgruntled system administrator uses a thumb drive to download and transfer a key logger to his supervisor's computer. The next day, using his boss's login credentials, he accesses the company's network, sends out a worrisome mass email that causes a lot of people to worry, and he promptly quits the company.

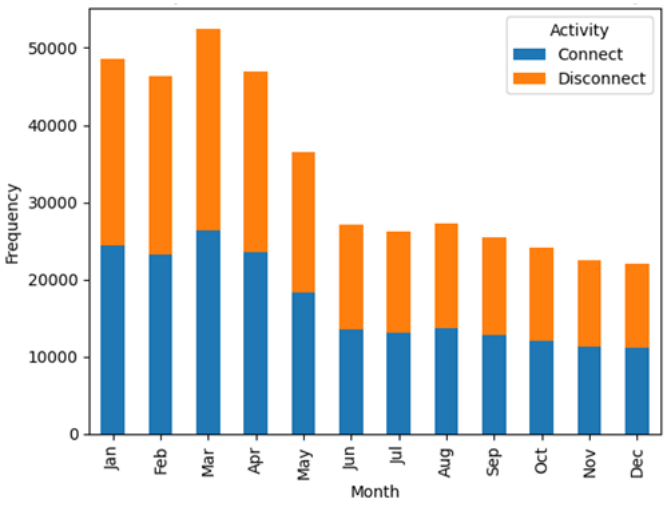


Fig. 4. Distribution of Device.csv

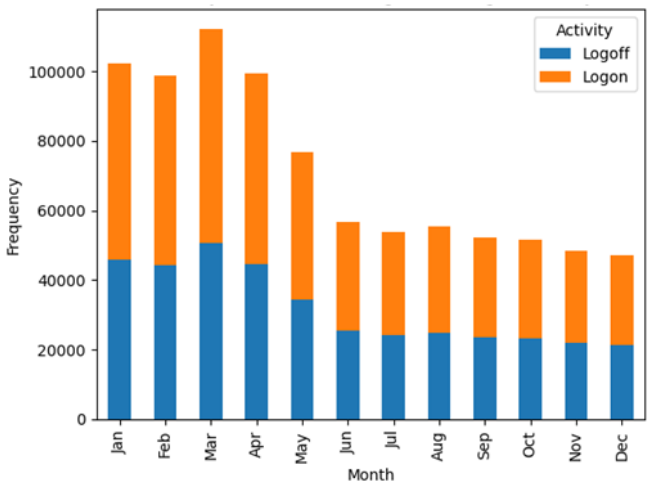


Fig. 5. Distribution of File.csv

### B. *Data pre-processing*

Data preparation is a crucial part in any research work. The data that is used in this work contains a large amount of user instance logs from various sources. All the data are not required for the work hence the relevant data in csv format were used and provided as input to the LSTM model for the training and testing process. Different activities of around 1000 users are provided in the dataset. Data is processed into smaller chunks where each chunk handles only the activities for the single users. With the repeated handling of this process, all the smaller chunks are then concatenated to produce the single input data for the training process of the data.

All the activities in the dataset contain a timestamp field. That timestamp field is converted into a standard date and time field and date and time field is used to analyze the user activity as malicious or normal. During the training process all the categorical and numerical columns were converted to tensors and can easily be provided as input during the training process.

The data in Cert 4.2 has all unlabeled data. So for supervised learning the data should be labeled first. For this purpose the activities of the user deviating from the normal office hour and normal days are labeled as possible threat and all other normal activities are labeled as normal.

Preprocessing includes cleansing, sampling, transformation and the result dataset was used as input for the model and finally normalization of the data to be fit in the model development is carried in the data processing stage [7]. Categorical data should be converted to the numerical form before feeding to the LSTM model for training and testing. Work is carried using label encoding.

### C. *Feature Selection*

The used data consists of unprocessed events or user actions. The extraction of features and the efficient correction of features during the feature engineering process constitute one of the main obstacles in anomaly detection. Since the suggested method relies on session activities, calculating user sessions is a crucial component of the study project. Since it is the first action a user performs, the login activity in the CERT dataset comes before the other actions. Subsequent actions related to the user are then carried out, and the session concludes when the user signs off the machine. Every user has several variable sessions connected to them during the day. The period of time between the log-in and log-off times is known as the user session. Depending on the use cases, a varying number of selected features is found in various study efforts. The two main categories of characteristics are categorical and numerical. The dataset's readme.txt file contains information that is used to choose numerical features [4].

Feature Extraction Example: Off hour activity is very important in this work. So, time between 8:00:00 and 19:00:00 is considered as normal working hours whereas rest of the time is considered to be off hour. If a user logon to the PC or connects a device and surfs a job site or hacking site in off hour then the user could be a probable insider threat.

For this work all the features present in all three files were relevant so all the feature were used except id that contains unique identifier for each instance.

Feature Extraction Process:

1. Merge Similar Rows
2. Time Window Definition
3. Aggregating data over time window, creating lag features.

Since dataset contains minimal number of features manual feature extraction was carried instead of some feature selection algorithm such as filtering methods, wrapper methods or dimensionality reduction techniques like PCA.

### D. *Model Development*

The LSTM based model is used to detect the presence of any insider activity within the organization. First the model is developed with different stages, this model is used to determine the user behavior as normal or abnormal. Different layers used are Sequential, DropOut and Dense, Activation function used is ReLu and Sigmoid.

The approach that is based on long short-term memory (LSTM) has been designed to identify instances of insider involvement in an organization. Tensorflow, a toolkit that offers robust support for the creation of several deep learning algorithms, has been utilized to implement the model. After the various stages of the model's development were finished, it was utilized to identify if the user activity data that was provided was indicative of normal or abnormal user behavior.

The LSTM class has been defined from the keras package. First a Sequential model was created in a linear stack of layers. The model was designed with multiple layers. The model have an LSTM layer followed by dropout, dense (fully connected) layers, and an output layer. An LSTM layer with 64 units was added. The input shape specifies the input size, which is 1 assuming a single feature or time series. The return sequences was “False” means that this LSTM layer does not return sequences (only the final output). Dropout helps prevent overfitting by randomly setting a fraction of input units to 0 during training. Dropout rate of 50% was used. A dense layer with 32 units was added and activation was ReLU. This layer introduces non-linearity and learns complex features. The final output layer has 1 unit with sigmoid activation (for binary classification).The model is compiled with the Adam optimizer and binary cross-entropy loss.The chosen metric for evaluation is accuracy.

Different layers of LSTM include:

- Sequential: each sequential layer corresponds to processing one time step of a sequence. The LSTM layer operates recursively for each time step, updating its internal state based on the current input and the state from the previous time step.
- ReLu: activation function.
- DropOut: used to prevent the overfitting.
- Sigmoid: The sigmoid function is used in each of gates to squish the input values into the range [0, 1].

### E. *Model Training*

This stage uses the training data to train the model where the features seen with training data are learnt by the model. Later these features are used by the model in the testing and making the prediction. Properly trained model provides reliable outcomes and those can be used for the practical implementation for the organization. Loss function and optimizer are defined to train the model. The model tries to deal with the classification problem due to which cross entropy loss along with the Adam optimizer are used in the model.

770000 data from the processed data have been used with the distribution of normal is to malicious data ratio of 1:10 which has reduced the over and under fitting the training. The training procedure was carried out across upto 80 epochs, and the loss function was used to calculate the training loss for each epoch. After adding the overall loss to each loss, the total loss during the training phase was eventually calculated. During the training phase, the gradient is also updated using the optimizer function. The Model class object was constructed in order to train the model. The model development process defined the model class. The trained model contains data such as the number of numerical columns, the size of the output (two in this example), the embedding size of the category columns, and the number of neurons in the hidden layer.

There have been three hidden layers taken into consideration, each with 64, 32, and 1 neurons. Given that there are only two possible outputs, the above model shows that the in_features value in the first linear layer is 32 and the out_features value in the last layer is 2. Prior to the model being trained, the loss function must be ascertained, and in order to do this, the optimizer and loss function that were utilized in the model's training must be defined. In the case of the optimizer function, Adam optimizer has been applied, and the cross entropy loss has been employed as the classification-based detection method.

With various experiments and result. LSTM model with parameter values in Table 3 have the optimum performance on the basis of Accuracy, Loss, Confusion matrix, ROC-AUC and PR-AUC.

TABLE I. LSTM MODEL HYPERPARAMETERS.

| Parameters | Value |
|---|---|
| Activation Function | ReLu |
| Optimizer | Adam |
| No. of Epoch | 80 |
| Loss Function | Binary Cross Entropy |
| Batch Size | 1024 |
| Drop Out | 50% |

### F. Making Prediction

After model creation and training, the next task is the testing of the data. Among all the data available 20 percent data is used for testing the model. Test data are passed through the LSTM model, the returned values are compared to the actual test data output from the model. Once the test process is complete, it is used to determine the loss of the model.

### G. Data Validation

Data balancing will be used in the original dataset to create the uniform distribution of normal cases and abnormal cases. The model prepared earlier is validated with the help of randomly selected 20 percent of the data from the test data. The validation of the detection system is evaluated with the help of confusion matrix, Accuracy, Precision, Recall and F1 score.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall: Recall (sometimes referred to as true positive rate or sensitivity) is the percentage of true positives among all genuinely positive cases. This statistic assesses the degree to which the actual positive cases were correctly predicted.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

F1- Score is the harmonic mean of precision and recall. F-beta score is the weighted harmonic mean of precision and recall with the optimal value at 1 and worst value at 0. The beta parameter signifies the ratio or recall importance to precision importance. The value of $\beta$ shows among recall and precision which is important. In case of this research recall is more important and hence FN assumes higher priority so $\text{F} - \beta score$ can calculate where $\beta > 1$.

$$\text{F} - \beta = \frac{(1 + \beta^2) \times Precision \times Recall}{(\beta^2) \times Precision + Recall} \tag{4}$$

When $\beta$ =2,

$$\text{F2} - \text{score} = \frac{5 \times Precision \times Recall}{4 \times Precision + Recall} \tag{5}$$

Receiver Operator Characteristic (ROC) curve is an evaluation metric in classification problem. At various threshold settings, it shows TPR vs FPR. The classifier's capacity for class distinction and ROC curve summarization is measured by the area under the curve, or AUC. The more the AUC value is greater, the more effective the model is at the classification task. The system is stronger the higher its AUC score. The main reason for its application is its capacity to examine the issue of class imbalance. The balanced classification does not require ROC/AUC analysis [19].

## IV. RESULT AND DISCUSSION

### A. Result using LSTM model

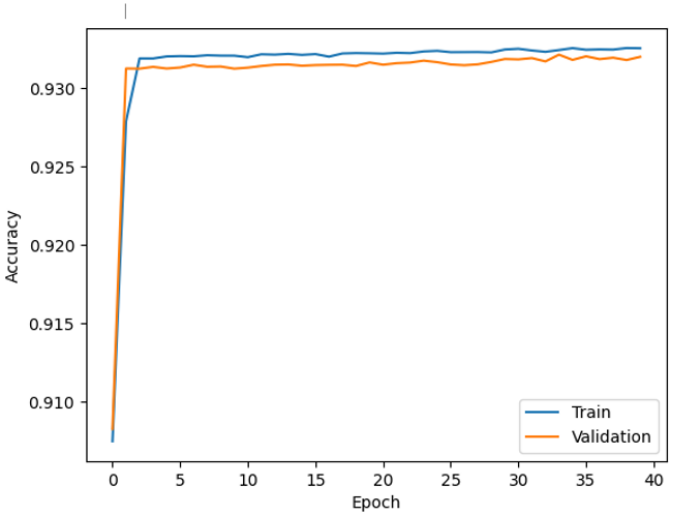


Fig. 6. Accuracy of the model

Using the system logs, deep learning algorithms have been utilized to assess the insider's threat. The LSTM-based algorithm have been used to anticipate if the information presented is harmful or legitimate data. The study attempted to comprehend the role that data distribution had in the training process, leading to an evaluation of the outcomes using the original dataset as many researches have tried to use the created dataset after processing the available data.

An LSTM layer accuracy of 70% was seen in the insider threat categorization [20]. However with the careful use of the system log files and features extraction and parameter tuning, the accuracy of the LSTM layer was observed to reach 93% without any synthetic data processing methods like GAN for senario 1. As a result, it was possible to increase the model's efficacy in threat prediction by distributing the data more uniformly.

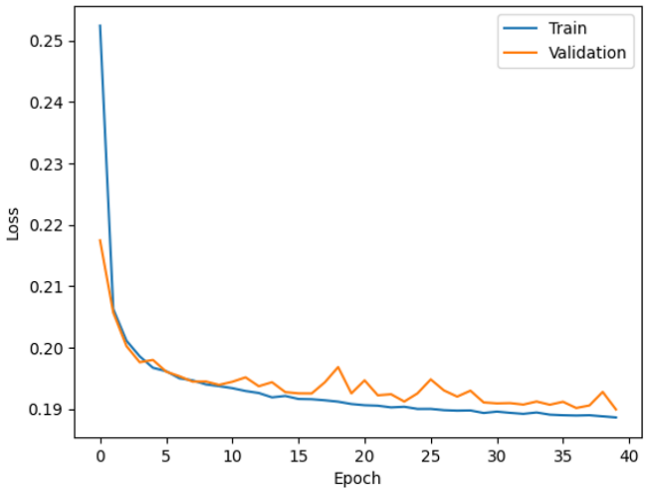


Fig. 7. Training Loss of the model

Initially, the data was classified using LSTM-based classification to identify if the activities were harmful or normal. Since the loss during training was seen to be consistent, 40 epochs were employed for the LSTM model's training procedure. Figure below shows the loss associated with the LSTM training and testing

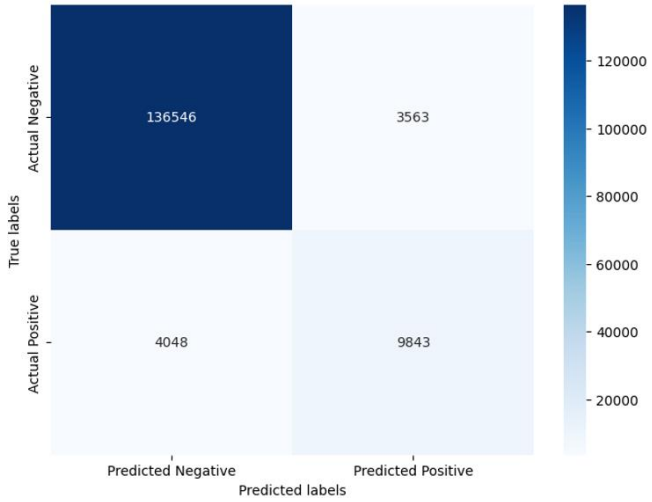


Fig. 8. Confusion Matrix

One of the crucial instruments for assessing and optimizing the categorization model is the ROC-AUC. Plotting True Positive Rate (TPR) or sensitivity on the y-axis against False Positive Rate (FPR) on the x-axis results in this curve. The trade-off between TPR and FPR is displayed via the ROC curve. Better performance will be indicated by the classifier whose curves are closer to the top-left corner. The diagonal line, where TPR and FPR are equal, will serve as the baseline for the random classifier. Therefore, a curve that approaches the ROC space's 45-degree diagonal indicates that the classification is less reliable.

The classifier model's performance can be ascertained by looking at the area under the ROC curve. The degree or measure of separation will be represented by this curve. A curve will be used to display the model's ability to differentiate between the classes. A higher AUC indicates that the model can predict the likelihood of a true class more accurately than a false class. AUC will have a value between 0 and 1, with 1 being a model that predicts with 100% accuracy

TABLE II. LSTM MODEL HYPERPARAMETERS.

| epoch | Input Neurons | Execution (ms/step) | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|---|---|
| 40 | 64 | 19 | 0.9314 | 0.6626 | 0.4891 | 0.5628 |
| 40 | 128 | 45 | 0.9318 | 0.6670 | 0.4890 | 0.5643 |
| 20 | 64 | 19 | 0.9314 | 0.6626 | 0.4885 | 0.5624 |
| 20 | 128 | 48 | 0.9312 | 0.6640 | 0.4812 | 0.5580 |

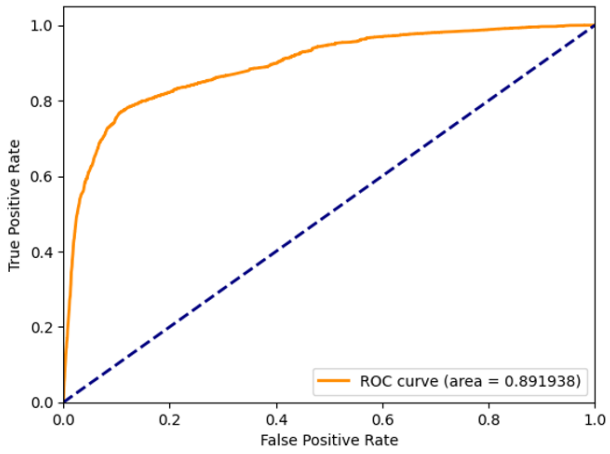


Fig. 9. ROC curve

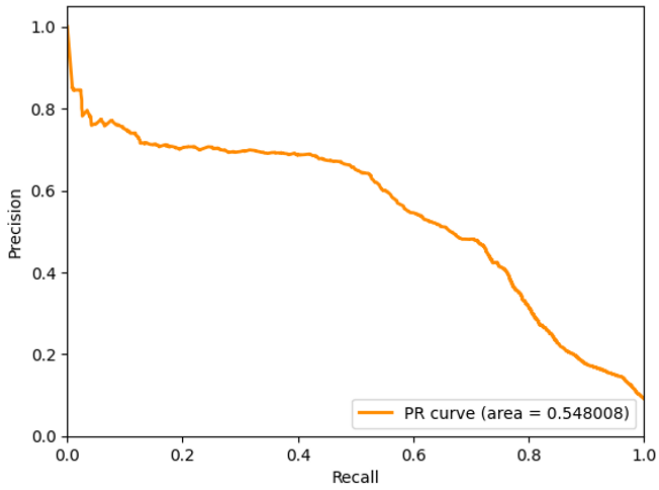


Fig. 10. PR-ROC Curve

## V. CONCLUSION AND RECOMMENDATION

### A. CONCLUSION

With the increased usage of digital platforms for service delivery, threats to the organizational network have become a significant challenge. The majority of enterprises have made the transition to digital platforms and are now keeping their data on digital networks. Any organization's data is a valuable asset since it serves as the foundation for many daily operations and long-term planning decisions. Attackers will constantly be looking for ways to exploit weaknesses so that they can manipulate important data in order to obtain access to it. Network-based attackers have used a variety of techniques to obtain access to organizational data, which has put organizations' security at serious risk. This work has attempted to examine the insider danger to the company, despite the fact that attackers based outside as well as inside the organization are present today. Because insiders are perceived as a significant threat to the business because of their position within it, this study has attempted to categorize whether insider activities are carried out with benign or malevolent intent.

The CMU CERT department has been managing and maintaining the dataset that was used for the thesis work with various parameters. While LSTM techniques were employed in this thesis work, deep learning algorithms were used to create the classification model. After the dataset was analyzed, it became clear that there was an imbalance in the distribution of abnormal and normal activity.

When using the produced dataset, the classification accuracy rose to 93% classification accuracy from the original dataset without any significant data preprocessing. It is thought that a significant contributing factor to the model's improved performance has been the choice of the relevant features and the choice of the training and test data within the dataset.

The organization will benefit from the deep learning-derived classification and detection model in terms of staff activity monitoring and data storage security. When any harmful activity is detected, the business can take action based on the outcome, enhancing the protection of organizational data from insider threats.

### B. RECOMMENDATION

As this work is primarily focused on developing a model for classifiying and detecting the insider threat of any organization without significant preprocessing the data. LSTM is seen promising for the purpose if feature extraction is done with great care. To even increase the accuracy and the precision of the work, for furthur enhancement, the hybrid model could be developed like with the use of LSTM and some other models like SVM, self attention layer might significantly increase the performance of the classification and detection system.

## REFERENCES


[1] M. Garuba, C. Liu, and D. Fraites, “Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems,” IEEE Xplore, Apr. 01, 2008

[2] Noever, D. Classifier Suites for Insider Threat Detection. arXiv 2019, arXiv: 1901.10948.

[3] Cappelli, D.; Moore, A.; Trzeciak, R. The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud); Addison-Wesley: Boston, MA, USA, 2012.

[4] Insider Threat Test Dataset, CMU University, In https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247

[5] Kulik, T.; Dongol, B.; Larsen, P.G.; Macedo, H.D.; Schneider, S.; Tran-Jørgensen, P.W.; Woodcock, J. A survey of practical formal methods for security. Form. Asp. Comput. 2022, 34, 1–39.

[6] Rauf, U.; Shehab, M.; Qamar, N.; Sameen, S. Formal approach to thwart against insider attacks: A bio-inspired auto-resilient policy regulation framework. Future Gener. Comput. Syst. 2021, 117, 412–425.

[7] Krichen, M.; Lahami, M.; Cheikhrouhou, O.; Alroobaea, R.; Maâlej, A.J. Security testing of internet of things for smart city applications: A formal approach. In Smart Infrastructure and Applications; Springer: Berlin/Heidelberg, Germany, 2020; pp. 629–653.

[8] Santosh Nepal, Basanta Joshi, 2021, In Proceedings of 10th IOE Graduate Conference; pp. 232-238

[9] Larsen, K.; Legay, A.; Nolte, G.; Schlüter, M.; Stoelinga, M.; Steffen, B. Formal Methods Meet Machine Learning (F3ML). In Proceedings of the Leveraging Applications of Formal Methods, Verification and Validation. Adaptation and Learning, Rhodes, Greece, 22–30 October 2022; Margaria, T., Steffen, B., Eds.; Springer Nature Switzerland: Cham, Switzerland, 2022; pp. 393–405.

[10] Urban, C.; Miné, A. A review of formal methods applied to machine learning. arXiv 2021, arXiv:2104.02466.

[11] Chen, H.; Zhang, H.; Si, S.; Li, Y.; Boning, D.; Hsieh, C.J. Robustness verification of tree-based models. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.

[12] Ranzato, F.; Zanella, M. Robustness verification of support vector machines. In Proceedings of the International Static Analysis Symposium, Porto, Portugal, 8–11 October 2019; Springer: Cham, Switzerland, 2019; pp. 271–295.

[13] Pantelidis, E.; Bendiab, G.; Shiaeles, S.; Kolokotronis, N. Insider Threat Detection using Deep Autoencoder and Variational Autoencoder Neural Networks. In Proceedings of the 2021 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece, 26–28 July 2021; pp. 129–134.

[14] Tuning the Hyper-Parameters of an Estimator. Available online: https://scikit-learn.org/stable/modules/grid_search.html (accessed on 17 May 2022).

[15] Insider Threat Test Dataset; Software Engineering Institute: Pittsburgh, PA, USA, 2016.

[16] Yuan, S.; Wu, X. Deep Learning for Insider Threat Detection: Review, Challenges and Opportunities. arXiv 2020, arXiv:2005.12433.

[17] Gavai, G.; Sricharan, K.; Gunning, D.; Hanley, J.; Singhal, M.; Rolleston, R. Supervised and Unsupervised methods to detect Insider Threat from Enterprise Social and Online Activity Data. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl. 2015, 6, 47–63. [CrossRef]

[18] Al-Mhiqani, M.N.; Ahmad, R.; Zainal Abidin, Z.; Yassin, W.; Hassan, A.; Abdulkareem, K.H.; Ali, N.S.; Yunos, Z. A Review of Insider Threat Detection: Classification, Machine Learning Techniques, Datasets, Open Challenges, and Recommendations. Appl. Sci. 2020, 10, 5208.

[19] Meng, F.; Lou, F.; Fu, Y.; Tian, Z. Deep Learning Based Attribute Classification Insider Threat Detection for Data Security. In Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, China, 18–21 June 2018; pp. 576–581. [CrossRef]

[20] Sagar Khanal (2022). Insider Threat Detection Using Deep Learning on System Log. Masters Thesis. Tribhuvan University , Institute of Engineering Pulchowk Campus

[21] Sharma, B.; Pokharel, P.; Joshi, B. User Behavior Analytics for Anomaly Detection Using LSTM Autoencoder–Insider Threat Detection. In Proceedings of the 11th International Conference on Advances in Information Technology, Bangkok, Thailand, 1–3 July 2020; ACM: Bangkok, Thailand, 2020; pp. 1–9.

[22] Orizio, R.; Vuppala, S.; Basagiannis, S.; Provan, G. Towards an Explainable Approach for Insider Threat Detection: Constraint Network Learning. In Proceedings of the 2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), San Antonio, TX, USA, 5–7 September 2022; pp. 42–49.

[23] Tian, Z.; Shi, W.; Tan, Z.; Qiu, J.; Sun, Y.; Jiang, F. Deep Learning and Dempster-Shafer Theory Based Insider Threat Detection. Mob. Netw. Appl. 2020.

[24] Haidar, D.; Gaber, M.M. Data Stream Clustering for Real-Time Anomaly Detection: An Application to Insider Threats. In Clustering Methods for Big Data Analytics: Techniques, Toolboxes and Applications; Nasraoui, O., Ben N'Cir, C.E., Eds.; Unsupervised and Semi-Supervised Learning; Springer International Publishing: Cham, Switzerland, 2019; pp. 115–144.

[25] Yuan, F.; Cao, Y.; Shang, Y.; Liu, Y.; Tan, J.; Fang, B. Insider Threat Detection with Deep Neural Network. In Proceedings of the ICCS, Wuxi, China, 11–13 June 2018.

[26] Raval, M.S.; Gandhi, R.; Chaudhary, S. Insider Threat Detection: Machine Learning Way. In Versatile Cybersecurity; Conti, M., Somani, G., Poovendran, R., Eds.; Advances in Information Security; Springer International Publishing: Cham, Switzerland, 2018; pp. 19–53.

[27] Malhotra, P.; Vig, L.; Shroff, G.M.; Agarwal, P. Long Short Term Memory Networks for Anomaly Detection in Time Series. In Proceedings of the ESANN, Bruges, Belgium, 22–23 April 2015.

[28] Kwon, D.; Natarajan, K.; Suh, S.C.; Kim, H.; Kim, J. An Empirical Study on Network Anomaly Detection Using Convolutional Neural Networks. In Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2–6 July 2018; pp. 1595–1598. ISSN 2575-8411.

[29] Koutsouvelis, V.; Shiaeles, S.; Ghita, B.; Bendiab, G. Detection of Insider Threats using Artificial Intelligence and Visualisation. In Proceedings of the 2020 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, 29 June–3 July 2020; pp. 437–443.

[30] Sheykhkanloo, N.M.; Hall, A. Insider Threat Detection Using Supervised Machine Learning Algorithms on an Extremely Imbalanced Dataset. Int. J. Cyber Warf. Terror. 2020, 10, 1–26. [CrossRef]

[31] Singh, M.; Mehtre, B.M.; Sangeetha, S. User Behavior Profiling using Ensemble Approach for Insider Threat Detection. In Proceedings of the 2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA), Hyderabad, India, 22–24 January 2019; pp. 1–8. ISSN 2640-0790.

[32] Wang, W.; Zhu, M.; Wang, J.; Zeng, X.; Yang, Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July 2017; pp. 43–48.

[33] Ren, Y.; Wu, Y. Convolutional deep belief networks for feature extraction of EEG signal. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 2850–2853. ISSN 2161-4407.

[34] Al-Mhiqani, M.N.; Ahmad, R.; Zainal Abidin, Z. An Integrated Imbalanced Learning and Deep Neural Network Model for Insider Threat Detection. Int. J. Adv. Comput. Sci. Appl. 2021, 12, 2021.

[35] Gayathri, R.G.; Sajjanhar, A.; Xiang, Y.; Ma, X. Multi-class Classification Based Anomaly Detection of Insider Activities. arXiv 2021, arXiv: 2102.07277.

[36] Mohammed, M.A.; Kadhem, S.M.; Maisa'a, A.A. Insider Attacker Detection Using Light Gradient Boosting Machine. TechKnowledge 2021, 1, 48–66.

[37] Singh, M.; Mehtre, B.M.; Sangeetha, S. Insider Threat Detection Based on User Behaviour Analysis. In Proceedings of the Machine Learning, Image Processing, Network Security and Data Sciences, Silchar, India, 30–31 July 2020; Bhattacharjee, A., Borgohain, S.K., Soni, B., Verma, G., Gao, X.Z., Eds.; Communications in Computer and Information Science; Springer: Singapore, 2020; pp. 559–574. [CrossRef]

[38] Rastogi, N.; Ma, Q. DANTE: Predicting Insider Threat using LSTM on system logs. arXiv 2021, arXiv: 2102.05600.

[39] Gayathri, G.R.; Sajjanhar, A.; Xiang, Y. Image-Based Feature Representation for Insider Threat Classification. arXiv 2019, arXiv: 1911.05879.

[40] Aldairi, M.; Karimi, L.; Joshi, J. A Trust Aware Unsupervised Learning Approach for Insider Threat Detection. In Proceedings of the 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), Los Angeles, CA, USA, July 30–1 August 2019; pp. 89–98.

[41] Kim, D.W.; Hong, S.S.; Han, M.M. A study on Classification of Insider threat using Markov Chain Model. KSII Trans. Internet Inf. Syst. 2018, 12, 1887–1898.

[42] Le, D.C.; Nur Zincir-Heywood, A. Machine learning based Insider Threat Modelling and Detection. In Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Washington, DC, USA, 8–12 April 2019; pp. 1–6. ISSN 1573-0077.