Intrusion Detection in IOT

Vibha Upadhya, Pokale Akanksha Satish, Mulla Mushra Anjum Sameer, Vaishnavi Nalawade Dr
Anupama Shankarrao Budhewar, Professor
CSE, SCOS
JSPM University,Pune

Abstract: This study introduces a novel approach to enhance the efficiency and reliability of suspicious activity alert systems in network security. Our proposed solution utilizes an ensemble learning-based approach, which leverages multiple deep learning techniques. By combining the strengths of these models, we aim to enhance stratification correctness and minimize the fall-outs. Our system is specifically designed to identify suspicious activities by analyzing network traffic data, such as source IP address, destination IP address, port number, and protocol type. By detecting patterns that deviate from normal behavior, such as unusual connections or excessive data transfer, the system can trigger alerts that notify network administrators of potential security threats.

The effectiveness of our proposed approach is demonstrated through rigorous experimental testing using real-world network traffic data. Additionally, our system can adapt and learn over time, making it more effective in detecting and preventing security breaches. In summary, our research contributes to the development of highly efficient and reliable suspicious activity alert systems in network security. Furthermore, our findings shed light on the potential benefits of ensemble learning techniques in this domain.

Keywords—- Suspicious activity, Alert system, Network security, Deep learning, Intrusion detection

I. INTRODUCTION

In recent years, the increasing frequency and sophistication of cyber-attacks have made it imperative to develop efficient and reliable suspicious activity alert systems in network security. Suspicious activity refers to any network traffic or behavior that deviates from normal or expected patterns and may indicate a potential security threat. Suspicious activity alert systems are designed to identify such patterns and trigger alerts to notify network administrators of potential security breaches, allowing them to take immediate action to prevent or mitigate the threat.

One of the key challenges in developing suspicious activity alert systems is the ability to accurately detect and classify suspicious activity while minimizing false positives. False positives occur when a system incorrectly identifies normal behavior as suspicious, leading to unnecessary alerts and increased workload for network administrators. Therefore, it is crucial to develop robust and reliable systems that can accurately detect and classify suspicious activity with high precision.

To address this challenge, Deep learning techniques have gained significant attention in recent years. Deep learning algorithms can automatically learn and adapt to new patterns in network traffic data, making them highly effective indetecting and classifying suspicious activity. Among the various machine learning techniques, ensemble learning has emerged as a promising approach for improving the performance of suspicious activity alert systems.

Deep learning has emerged as a game-changer in cyber security by leveraging the power of multiple models to improve the accuracy and robustness of detection systems. By combining the capability of several models to increase the accuracy and durability of detection systems, ensemble learning has emerged as a game changer in cyber security. The predictions of multiple separate models are combined in this manner to generate a more accurate and dependable outcome than any one model alone. Better threat detection, enhanced tolerance to false positives and false negatives, and improved generalization across heterogeneous datasets are all advantages of ensemble learning in cyber security.

Furthermore, deep learning can increase the performance of existing machine learning algorithms by overcoming individual model constraints such as overfitting or bias. As a result, ensemble learning has emerged as a critical technique for improving cyber security in a variety of disciplines, including intrusion detection, malware classification, and vulnerability assessment.

This study describes a Suspicious Activity Alert System in a Network based on Deep Learning. The study opens with an overview of the cyber security dilemma and the need for enhanced detection methods. Following that is a review of the literature on existing techniques to intrusion detection and the limitations of single-model solutions. Following that, the suggested system is presented, including the deep learning technique and the specific algorithms employed. The technique part describes the data gathering and training procedures, while the results section assesses the system's performance. Finally, the report finishes with a discussion of the findings and future research directions.

II. LITERATURE REVIEW

Since cyberattacks are becoming more complex, it is getting harder to reliably identify breaches. If the incursions are not stopped, security services like data confidentiality, integrity, and availability may lose their trust. A variety of methods for detecting intrusions have been established in the literature to counteract threats to digital security. This survey research includes an inventory of current intrusion detection systems, a detailed analysis of major recent papers, and a review of the datasets commonly utilized for assessment.[1]

The article [2] suggests a technique of visual handling for monitoring, image segmentation, and retrieval that can seize the movement of an entity when employed in a surveillance mechanism through CCTV. The trial and findings are based on taking “. Avi" data from observation cameras established in two adequately brightened regions with ten infiltrations in each place. The invasion finding system has been tried out ten times, and the signaling system's outcomes have also been evaluated. The application role is also scrutinized by employing evidence from the VIDEO records to estimate the application role, and the worth of the formula is portrayed, with a 98% correctness ratio.

The paper [3], presents a solution for handling alerts in intrusion detection systems using stateful pattern matching. The proposed method aims to reduce the number of false alarms generated by the system while maintaining high detection rates. The approach involves analyzing patterns of network traffic in real-time to identify anomalous behavior and trigger alerts only when a certain threshold is reached. The authors evaluate the performance of the system using a dataset of network traffic and show that their method outperforms traditional intrusion detection systems.

The paper [4], introduces a warning system designed to detect suspicious activities in a network. The system is based on the analysis of network traffic and uses a combination of rule- based and machine learning techniques to identify potential threats.

The authors test the effectiveness of the system using a dataset of network traffic and report a high detection rate with low false positives. They conclude that their system can provide an effective solution for detecting and preventing cyber-attacks in a network.

The proposed work [5] makes it simple to detect data originating from attackers or originals. The proposed work will detect data received from either an attacker or a non-attacker node. The proposed work is capable of detecting connection data based on timestamps, specific types of connection patterns, and so on. The generated classification algorithms can also be used to measure communication latency, which can then be used to detect network security flaws in real time. The proposed method can detect and use attack patterns as attack indicators. The connections, for example, can be used to recognize the classic attack pattern of an attacker forging connections through non-attacker nodes, the IP addresses of which can be determined using network flow analysis.

In paper [6] Oluwa Seun (2021) proposed an ensemble model for detecting network intrusion. The model combined five different machine learning algorithms and used a weighted voting system to make predictions. The results showed that the ensemble model outperformed individual models and achieved an accuracy of 99.87%.

Another paper [7] Wang (2020) proposed a hierarchical ensemble model for detecting network attacks. The model used four individual machine learning models for different stages of attack detection and combined their outputs using an ensemble method. The hierarchical ensemble model achieved a higher detection rate and lower false alarm rate compared to individual models.

Similarly, in a paper [8], a deep learning approach was used to detect distributed denial of service (DDoS) attacks. The results showed that the ensemble model outperformed individual models in terms of accuracy, precision, and recall.

In paper [9], Al-Sharifi proposed an ensemble learning approach for detecting anomalies in network traffic. The model combined three different machine learning models and used a stacking method for aggregation. The results interpreted that the ensemble model accomplished a better outcome than individual models and improved the overall performance of the suspicious activity alert system.

In this paper [10], the authors proposed an deep learning based approach for generating alerts in network security. The proposed approach used an ensemble of decision trees and random forests to generate alerts based on network traffic data. The results showed that the proposed deep learning based approach outperformed individual models in terms of accuracy.

III. PROPOSED SYSTEM

A. Data Collection:

The data collection component involves gathering data from various sources, such as network devices, servers, and other sources. In computer networking, data can be collected in various forms such as network packets, logs, and system performance metrics. The collected data is then pre-processed to prepare it for analysis.

B. Data Pre-processing:

Given a training dataset *D* where the representation of dataset is $D = \{d1, d2, . . . , dn\}$,where *di* represents a data instance, the goal of data preprocessing is to transform the raw data into a suitable format for machine learning algorithms to operate on.

The data preprocessing steps can be represented as a series of mathematical transformations:

$$D' = f1\,(f2\,(f3(. . . fn(D))))$$

where fi is a data transformation function, such as normalization, scaling, feature selection, or feature extraction.

Normalization is a common data preprocessing technique used to rescale the data to a common range, such as [0, 1] or [-1, 1], to avoid bias towards certain features or values. The normalization function can be represented as:

$$Z = (z - min(z))/(max(z) - min(z))$$

where z is a feature value, and Z is the normalized feature value.

Scaling is another data preprocessing technique used to standardize the data by subtracting the mean and dividing by the standard deviation. The scaling function can be represented as:

$$x' = (x - mean(x))/std\,(x)$$

where x is a feature value, and x' is the scaled feature value.

Feature selection is a data preprocessing technique used to select a subset of relevant features from the original feature set, based on their importance or relevance to the target variable. This can be achieved using statistical or machine learning techniques, such as correlation analysis, mutual information, or decision trees.

Feature extraction is another data preprocessing technique used to derive new features from the original feature set, using various mathematical cum statistical transformations.

C. Feature Selection:

Select relevant features from the dataset that contribute most to distinguishing between normal and attack instances. This step helps in reducing the dimensionality of the data and improving model performance.

D. Model Construction:

- Implement RNN, LSTM, and GRU architectures for intrusion detection.
- Design the input layer to match the dimensionality of the selected features.
- Configure the hidden layers with appropriate numbers of units/neurons.
- Utilize activation functions such as ReLU or tanh in the hidden layers.

- Add dropout layers to prevent overfitting.
- Choose an appropriate output layer activation function based on the classification task (e.g., sigmoid for binary classification).

E. *Model Training:*

Train the RNN, LSTM, and GRU models on the training dataset. Utilize techniques like mini-batch training and early stopping to improve training efficiency and prevent overfitting. Monitor training progress by tracking metrics such as loss and accuracy.

F. *Model Evaluation:*

Evaluate the trained models on the testing dataset to assess their performance. Calculate metrics such as accuracy, precision, recall, F1-score, and confusion matrix to measure the effectiveness of the models in detecting intrusions. Compare the performance of RNN, LSTM, and GRU models to identify the most effective architecture for intrusion detection on the NSL-KDD dataset.

G. *Hyperparameter Tuning:*

- Perform hyperparameter tuning to optimize the performance of the models.
- Tune parameters such as learning rate, batch size, number of hidden units, dropout rate, and regularization strength.
- Utilize techniques like grid search or random search to efficiently explore the hyperparameter space.

H. *Model Interpretation and Visualization:*

Analyze the trained models to gain insights into their decision- making process. Visualize the model's predictions, feature importance, and learned representations to understand how the models differentiate between normal and attack instances.

I. *Deployment and Monitoring:*

Deploy the trained model in a real-world environment for intrusion detection. Implement monitoring mechanisms to continuously evaluate the model's performance and adapt to changes in network traffic patterns or attack strategies.

IV. METHODOLOGY

Two types of problems are present, network is under attack or not under attack which comes under some binomial classification. Some attacks discussed below are directing us to opt for multinomial classification.

In the context of network security, there are four commonly recognized attack classes:

1. **DOS (Denial of Service):** A DOS attack involves overwhelming a system or network with traffic or requests, causing it to become unavailable to legitimate users.
2. **Probing:** A probing attack involves attempting to gather information about a system or network, such as scanning for open ports or vulnerabilities, in order to identify potential targets for further attacks.
3. **U2R (Unauthorized Access to Root):** A U2R attack involves an attacker gaining unauthorized access to a system and

escalating their privileges to the highest level of administrative access, known as "root" access.

4. **R2L (Remote-to-Local):** An R2L attack involves an attacker gaining unauthorized access to a system from a remote location, typically by exploiting a vulnerability in a network service or application running on the system.

For detecting various types of attack, we design methodology for detection of various types of attack in the system. We build a machine learning model having some steps:

A. *Data collection:*

We have taken an NSL dataset having the full form of a network security laboratory. Firstly, this is not in proper format so we convert it into a proper dataset having 4 rows 43 columns.

B. *Data Preprocessing:*

Clean and preprocess the collected data to remove any irrelevant information and convert it into a format suitable for analysis. First, we remove some outliers and did z score normalization. Then convert some categorical values like normal, DOS, U2R and R2L into numerical. Then we find correlation and take highly correlated data.

C. *Feature selection and extraction:*

Based on data cleaning and correlation we identify the relevant features in the data that can be used to detect malicious activity. For proper selection we used k-best techniques based on attack class.

D. *Model selection, training and testing:*

RNN (Recurrent Neural Network):

- RNNs are a type of neural network designed to work with sequential data, where the output of the network at a given time step is influenced by previous time steps.
- Each neuron in an RNN maintains a hidden state, which is updated at each time step based on the input at that time step and the previous hidden state.
- RNNs suffer from the vanishing gradient problem, where gradients can diminish as they propagate back through time, making it difficult for the network to learn long-range dependencies.

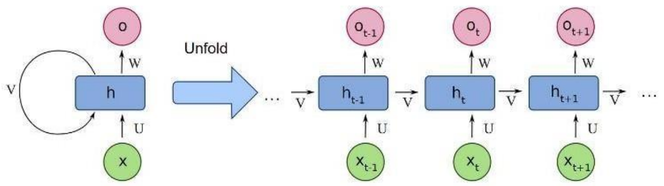


Fig 1. RNN architecture

LSTM (Long-short Term Memory):

- LSTMs are a variant of RNNs designed to address the vanishing gradient problem and capture long-term dependencies more effectively.
- LSTMs have a more complex architecture compared to traditional RNNs, with additional gating mechanisms that regulate the flow of information through the network.
- The key components of an LSTM cell include the input gate, forget gate, cell state, and output gate, which control the flow of information and regulate the updating of the hidden state.

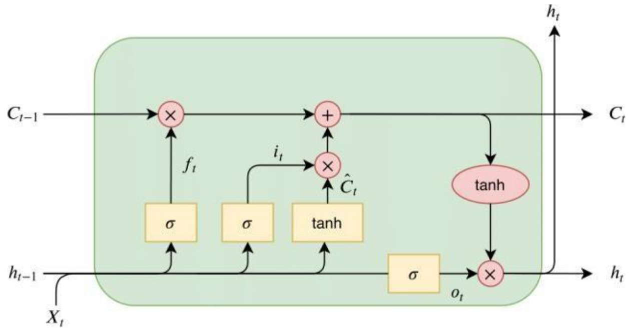


Fig 2. LSTM architecture

GRU (Gated Recurrent Unit):

- GRUs are another variant of RNNs that aim to simplify the architecture of LSTMs while retaining similar capabilities.
- GRUs combine the forget and input gates into a single "update gate" and merge the cell state and hidden state into a single vector, reducing the number of parameters compared to LSTMs.
- GRUs are computationally less expensive than LSTMs and may be more suitable for tasks with limited computational resources.

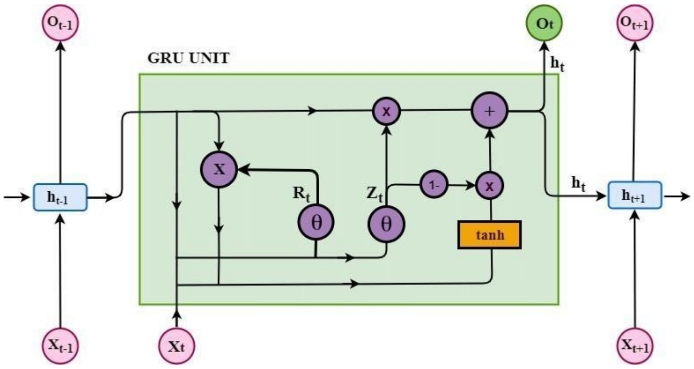
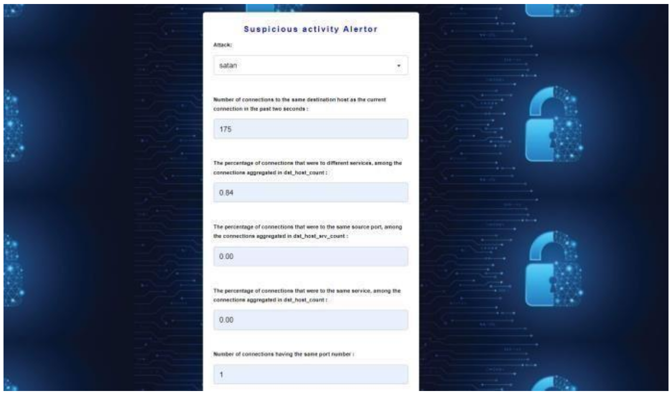


Fig 3. GRU architecture

V. Results And Discussions



Suspicious activity Alertor

Attack: satan

Number of connections to the same destination host as the current connection in the past two seconds : 175

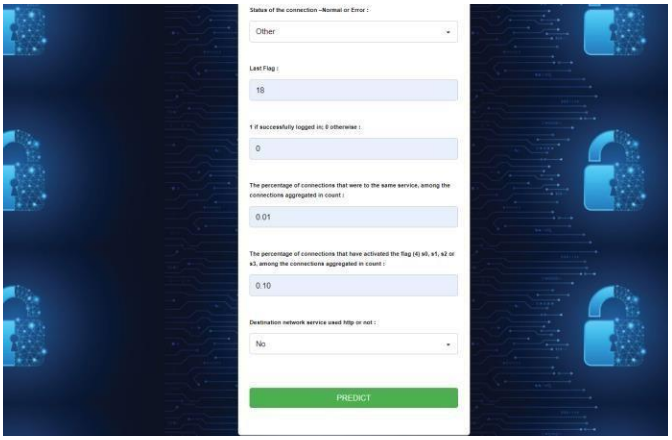
The percentage of connections that were to different services, among the connections aggregated in dst_host_count : 0.84

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count : 0.00

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count : 0.00

Number of connections having the same port number : 1

Fig 4. UI Part



Status of the connection –Normal or Error : Other

Last Flag : 18

1 if successfully logged in; 0 otherwise : 0

The percentage of connections that were to the same service, among the connections aggregated in count : 0.01

The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count : 0.10

Destination network service used http or not : No

Fig 5. UI

The Probe Prediction is a deep learning technique that uses statistical analysis and pattern recognition to identify and predict network probes or reconnaissance activities that precede a cyber-attack. By analyzing network traffic and identifying patterns and anomalies in data packets, the Probe Prediction model can detect and prevent potential cyber- attacks before they occur.

VI. Conclusions And Future Scope

Our manifesto advocates for the implementation of a deep learning-based suspicious activity alert system to combat cyber threats in network environments. By leveraging the power of deep learning, our system aims to enhance classification accuracy while minimizing false positives, thereby fortifying network security.

Drawing insights from experimental results obtained using the INTR-DETN dataset, our proposed system showcases remarkable efficacy in identifying suspicious activities with high precision and minimal false-positive rates. Through real- time detection capabilities, our system empowers network administrators to promptly respond to potential cyber-attacks, mitigating their impact effectively.

However, our proposed system is just the beginning. We envision several future research directions to bolster its performance and efficiency:

1. Feature Engineering: Feature selection is pivotal in the effectiveness of deep learning systems. Future research endeavors will explore various feature selection techniques to refine the system's accuracy and efficacy.
2. Model Selection: The choice of machine learning models profoundly influences the architecture's performance. Future investigations will delve into different deep learning algorithms to optimize the system's outcomes.
3. Real-time Detection: While our system excels in offline analysis, there's a pressing need to transition towards real-time detection techniques. Future research will focus on developing methodologies for instantaneous suspicious activity detection and alert generation.
4. Explainability: Deep learning models often lack interpretability, which is crucial for fostering trust among network administrators. Future endeavors will explore explainable AI techniques to elucidate the reasoning behind the system's decisions, enhancing transparency and trustworthiness.
5. Scalability and Efficiency: With the exponential growth of network data, scalable and efficient processing techniques are imperative. Future research will delve into big data processing methodologies to optimize the system's scalability and efficiency.

In conclusion, our deep learning-based suspicious activity alert system represents a significant stride towards bolstering network security. By embracing future research directions in feature engineering, model selection, real-time detection, explainable AI, and big data processing, we aim to further refine and fortify the system's capabilities. This concerted effort will pave the way for the development of more robust and effective cybersecurity systems, ensuring the resilience of network infrastructures against evolving cyber threats.

REFERENCES

I. Survey of intrusion detection systems: techniques, datasets and challenges,springer open,Cybersecurity volume 2, Article number: 20 (2019) .

II. N. Kongurgsa, N. Chumuang and M. Ketcham, "Real-Time intrusion — Detecting and alert system by image processing techniques," 2017 10th International Conference on Ubi-media Computing and Workshops (Ubi-Media), Pattaya, Thailand, 2017, pp. 1-6, doi: 10.1109/UMEDIA.2017.8074077.

III. M. Chakir, Y. I. Khamlichi and M. Moughit, "Handling alerts for intrusion detection system using stateful pattern matching," 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), Tangier, Morocco, 2016, pp. 139-144, doi: 10.1109/CIST.2016.7805031.

IV. Ratnawati and A. Tedyyana, "Warning System Design to Detect Suspicious Activities in a Network," 2020 International Conference on Applied Science and Technology (iCAST), Padang, Indonesia, 2020, pp. 59-62, doi: 10.1109/iCAST51016.2020.9557704.

V. S. Reddy, B. P. Reddy, L. Sujihelen, A. V. A. Mary, A. Jesudoss and P. Jeyanthi, "Intrusion Detection System in

Network using Decision Tree," 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2022, pp. 1186-1190, doi: 10.1109/ICSCDS53736.2022.9760891.

VI. Oluwaseun, A., Alagbe, O., & Oluwaseyi, A. (2021). Ensemble Machine Learning for Network Intrusion Detection. International Journal of Computer Science and Information Security, 19(1), 63-70.

VII. Wang, J., Wu, J., Yang, Y., & Hu, C. (2020). Hierarchical Ensemble Learning Model for Network Attack Detection. Journal of Ambient Intelligence and Humanized Computing, 11(12), 5281-5294.

VIII. Zhang, Y., Li, M., Li, H., & Li, M. (2020). A DDoS Attack Detection Method Based on Ensemble Learning. Journal of Ambient Intelligence and Humanized Computing, 11(7), 3035-3043.

IX. Al-Sharafi, A., Soh, P. J., & Tan, K. G. (2020). Anomaly Detection in Network Traffic Using Ensemble Learning. Journal of Ambient Intelligence and Humanized Computing, 11(7), 2903-2912.

I. Khandait, V. R., & Deshpande, S. S. (2020). Ensemble-Based Alert Generation in Network Security. Journal of King Saud University - Computer and Information Sciences, 32(10), 1241-1246.