Network Security with LSTM-powered Irregularity Detection

Author 1	**Author 2**
Laasya Vajjala	Vadapalli Surya
11848603	11861342

Abstract

This project addresses the critical challenge of anomaly detection in network traffic, motivated by the escalating cybersecurity threats in digital ecosystems. With the stream of sophisticated attacks, such as distributed denial-of-service (DDoS) assaults and advanced persistent threats (APTs), conventional methods struggle to effectively discern inconsistent patterns. Leveraging Long Short-Term Memory (LSTM) deep learning, our project aims to enhance anomaly detection accuracy by capturing intricate temporal dependencies within network traffic data. The unique advantage lies in the model's ability to discern subtle deviations from normal actions, providing robust protection against unusual and evolving threats. The project involves preprocessing raw network data, training LSTM models, and validating their efficacy in distinguishing irregularities. Additionally, we plan to optimize model parameters to ensure real-time applicability and scalability, fostering a proactive approach to cybersecurity in dynamically changing environments.

Introduction

The ever-evolving landscape of cybersecurity threats necessitates a robust approach to anomaly detection in network traffic. Recent incidents underscore the urgency of such initiatives, such as the discovery of Microsoft Azure SSRF vulnerabilities in October 2023, exposing potential code execution risks. Similarly, the Slack GitHub account hack in September 2023 accentuates the importance of stringent access controls and heightened security awareness among employees. Additionally, the persistent challenge of data breaches, as witnessed in major incidents affecting Deezer, Twitter, and WordPress plugins in 2023, reinforces the critical need for advanced anomaly detection systems to safeguard against unauthorized access and exfiltration.

Our project lies in its utilization of Long Short-Term Memory (LSTM) deep learning, enabling the model to capture intricate temporal dependencies within network traffic data. This empowers the system to discern subtle anomalies indicative of sophisticated attacks that conventional methods might overlook. Unlike rule-based systems, LSTM models can adapt and learn from evolving threats, providing a proactive defence mechanism. The temporal awareness of LSTMs enhances the accuracy of anomaly detection by considering the sequential nature of network activities, making it a potent tool for addressing the dynamic and complex nature of modern cybersecurity challenges.

The technical plan of this project involves several key steps. Firstly, raw network data will undergo preprocessing to extract relevant features and normalize the dataset. Subsequently, LSTM models will be trained on the pre-processed data, utilizing their ability to capture temporal dependencies for effective anomaly detection. Model validation will be performed on labelled datasets, ensuring robust performance. Hyperparameter tuning and optimization will follow to enhance real-time applicability and scalability. The incorporation of concepts such as recurrent neural networks (RNNs) and LSTM architectures underscores the project's commitment to leveraging cutting-edge deep learning techniques for network security. Additionally, continuous monitoring and adaptation of the model will be emphasized to address the evolving nature of cybersecurity threats, fostering a comprehensive and adaptive anomaly detection system.

Literature Review

The literature surrounding anomaly detection in network traffic provides a foundational understanding of the challenges and approaches within the cybersecurity domain. Noteworthy studies, such as "Deep Learning for Anomaly Detection: A Survey" by Chalapathy et al., outline the evolution of deep learning techniques in this context. The survey emphasizes the significance of recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks for capturing temporal dependencies in sequential data, laying the groundwork for the project's adoption of LSTM deep learning for enhanced anomaly detection. Additionally, works like "A Survey of Anomaly Detection Techniques in Network Traffic" by Patel et al. shed light on the diversity of approaches, from statistical methods to machine learning-based models, underscoring the necessity for advanced techniques to combat evolving cybersecurity threats.

Techniques inspired by related studies form a crucial aspect of the literature review. Research such as "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery" by Schlegl et al. introduces the concept of using generative adversarial networks (GANs) for anomaly detection, inspiring an exploration of complementary techniques in our project. Furthermore, the incorporation of concepts from "Intrusion Detection in IoT-Based Healthcare Systems: A Review" by Kumar et al. provides insights into the unique challenges posed by specific environments, driving the need for adaptive and specialized anomaly detection mechanisms.

The inspiration for this project is rooted in real-world incidents, exemplified by the Microsoft Azure SSRF Vulnerabilities discovered in October 2023. Exploiting Server-Side Request Forgery (SSRF) vulnerabilities, attackers could execute arbitrary code on affected systems, emphasizing the critical importance of timely detection and mitigation. This incident serves as a poignant reminder of the dynamic and evolving nature of cybersecurity threats, prompting the adoption of advanced anomaly detection techniques like LSTM deep learning. The project seeks to address the gaps highlighted by such vulnerabilities, aiming to fortify network security by effectively identifying and thwarting anomalous activities indicative of potential cyber threats.

Technical Plan

There are several key steps, each contributing to the robustness and effectiveness of the LSTM model. Below is a detailed elaboration of the techniques and a flowchart demonstrating the sequential steps:

1. Data Preprocessing:

   The first step is the preprocessing of raw network data, where relevant features are extracted, and the dataset is normalized. This process ensures that the data is in a suitable format for input into the LSTM model. Tools such as Wireshark and Bro may be employed for packet capturing and network data extraction, facilitating comprehensive data preprocessing.

2. LSTM Model Architecture:

   Subsequently, the core of the project revolves around training LSTM models on the pre-processed network data. LSTM networks, known for their ability to capture long-term dependencies in sequential data, are particularly well-suited for analysing the temporal aspects of network traffic. Python, with libraries like TensorFlow or PyTorch, will likely be utilized for the implementation of the deep learning model. These frameworks offer a flexible environment for building and training neural networks, providing the necessary tools for fine-tuning model parameters.

3. Model Validation:

   Once the LSTM model is trained, validation becomes a critical step in ensuring its efficacy. Labelled datasets, representative of both normal and anomalous network behaviour, will be employed to assess the model's ability to accurately distinguish anomalies.

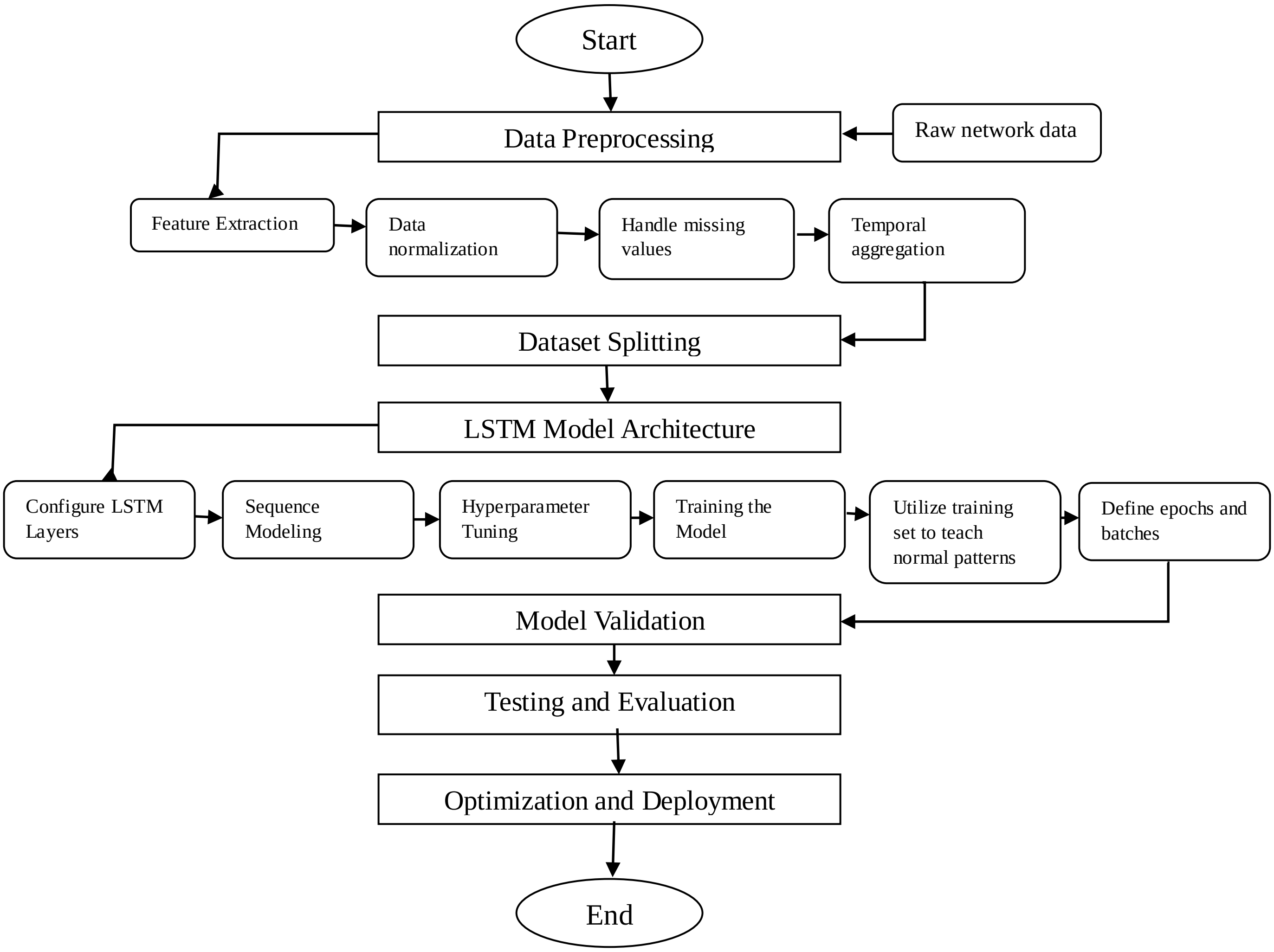
4. Testing and Evaluation:

   Evaluation metrics such as precision, recall, and F1 score will be utilized to quantify the model's performance. Additionally, visualization tools like Matplotlib or Seaborn may be employed to generate informative plots depicting model predictions and anomalies in the network traffic data.

5. Optimization and Deployment:

   To optimize the model for real-time applicability and scalability, hyperparameter tuning becomes overbearing. Techniques such as grid search or random search may be employed to systematically explore the hyperparameter space and identify configurations that enhance the model's overall performance. Continuous monitoring and adaptation mechanisms will be integrated, allowing the model to evolve and adapt to emerging cybersecurity threats dynamically.

Flowchart



References

1. Chalapathy, R., Chawla, S., & Robinson, P. (2019). Deep Learning for Anomaly Detection: A Survey. arXiv preprint arXiv:1901.03407.
2. Akoglu, L., Tong, H. & Koutra, D. (2015), 'Graph based anomaly detection and description: A survey', Data Mining and Knowledge Discovery.
3. Barreyre, N. (2011), 'The politics of economic crises: The panic of 1873, the end of reconstruction, and the realignment of American politics', The Journal of the Gilded Age and Progressive Era.

4. Kandanaarachchi, S. & Hyndman, R. J. (2021), ‘Dimension Reduction for Outlier Detection Using DOBIN’, Journal of Computational and Graphical Statistics
URL: https://doi.org/10.1080/10618600.2020.1807353
5. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q. Z., Xiong, H. & Akoglu, L. (2021), ‘A comprehensive survey on graph anomaly detection with deep learning’, IEEE Transactions on Knowledge and Data Engineering.
6. Tsikerdekis, M., Waldron, S. & Emanuelson, A. (2021), ‘Network anomaly detection using exponential random graph models and autoregressive moving average’, IEEE Access.