Enhancing Network Security with AI/ML Algorithms Along with Cloud Deployment

Apurba Chatterjee
System Software Engineer, Ericsson

***Abstract*— Early adopters of machine learning-powered network security solutions can gain a competitive edge by demonstrating a proactive and robust security posture. Machine learning can detect never-before-seen attacks (zero-day exploits) by recognizing unusual behavior, significantly enhancing a company's security posture. The KDD99 dataset is used as a base for the research. Network intrusion detection systems analyze network traffic to identify malicious activities. The activity for detection includes denial-of-service attacks, port scans, malware distribution, and unauthorized access attempts. Some of the algorithms that has been used while writing the python code are: XGBoost, LSTM and CNN. This will be an effective approach to any network-based systems. We can enact the algorithm deployment as a part of our daily cloud deployment.**

***Index Terms*—Machine Learning, malware, algorithms, XGBoost, LSTM, CNN, cloud.**

I. INTRODUCTION

ML can automate many tedious security tasks, freeing up security professionals to focus on more strategic initiatives. Early adopters of machine learning-powered network security solutions can gain a competitive edge by demonstrating a proactive and robust security posture. ML can analyze encrypted traffic without decryption, identifying malicious patterns and detecting threats hidden within encrypted communications. Like a sniffer dog trained to detect a specific scent, some systems rely on a database of known threats (signatures). They compare incoming network traffic or files against this database. If there's a match, it flags the threat. However, in case of the algorithms as discussed here, like a security guard who spots suspicious activity, machine learning models are trained on normal network behavior. They learn to recognize patterns and can identify deviations from the norm, even if they've never encountered that specific threat before.

II. PRE-REQUISITE KNOWLEDGES

Understanding of operating systems, networking concepts (TCP/IP, ports, firewalls), and intermediate level programming (Python is most recommended). A solid grasp of algebra, calculus, probability, and statistics is essential for understanding the underlying principles of the ML algorithms. KDD99 is typically used for supervised learning tasks (classification). One should understand concepts like classification, regression, training, testing, and evaluation metrics (accuracy, precision, recall, F1-score). Familiarity with algorithms like decision trees, support vector machines (SVMs), naive Bayes, k-nearest neighbors (KNN), and ensemble methods (e.g., random forests). Python is the go-to language for machine learning. Libraries like scikit-learn (for ML algorithms), pandas (for data manipulation), and NumPy (for numerical operations) are essential. Moreover, libraries like Matplotlib and Seaborn has helped to visualize data and gain insights.

III. IMPLEMENTATION

After importing all the necessary libraries, the dataset's features were defined, and the data was loaded. To ensure data quality, missing values were identified, with plans for handling them through imputation or removal as needed. Duplicate rows, which could potentially distort analysis and model training, were also eliminated. Subsequently, the correlation between numerical features in the KDD99 dataset was analyzed and visualized using a heatmap to provide insights into feature relationships. Next is creating a pair plot to visualize the relationships between four specific features in the KDD99 dataset and how these relationships vary across different intrusion types.

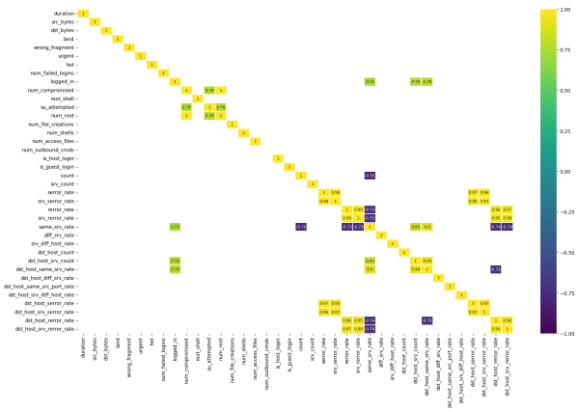


Figure-1 (Heatmap of the Dataset)

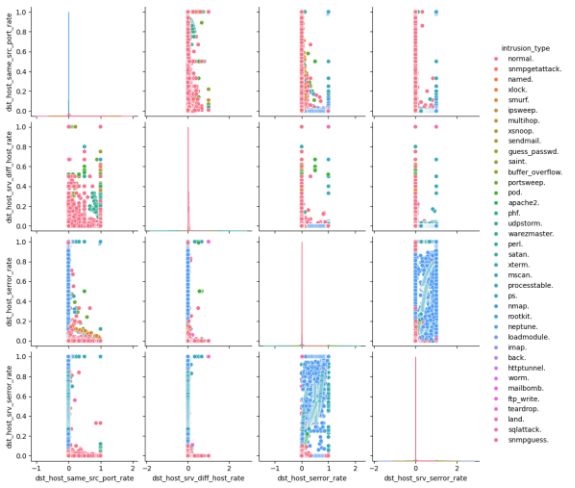


Figure-2 (The Pair Plot)

Now we used t-SNE (t-distributed Stochastic Neighbor Embedding) to reduce the dimensionality of the dataset and visualize it in a 2D space. It takes a subset of the data (10,000 samples each for 'normal' and 'neptune' attack types) to improve performance. It applies the t-SNE algorithm to reduce the dataset to two dimensions (n_components = 2). This helps visualize high-dimensional data by preserving the relationships between data points in a lower-dimensional space. In the next part of our coding, we visualized how the different intrusion types are clustered or separated in the 2D space created by t-SNE. This helped us in understanding if t-SNE effectively captured the underlying structure of the data and if different attack types can be visually distinguished. It uses seaborn's FacetGrid to create a scatter plot where each point represents a network connection.

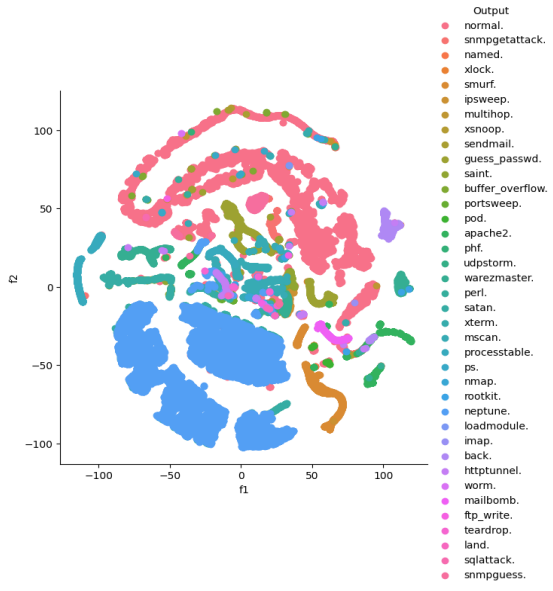


Figure-3 (Scatter Plot, 2D data generated by t-SNE)

Next steps were preparing the target variable, splitting the data, evaluating model performance using a confusion matrix, and calculating accuracy. Now the following steps were implemented which marked as the starting of the applications of the algorithm. Scaling and reducing the dimensionality of numerical features, One-hot encoding categorical features, combining all features into a single DataFrame, Adding the target variable, Saving the processed data, these preprocessing ensures that the data is in a suitable format for training the XGBoost model.

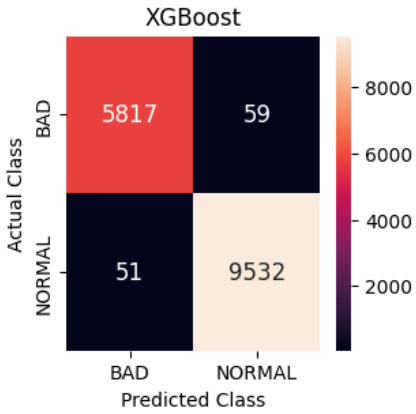
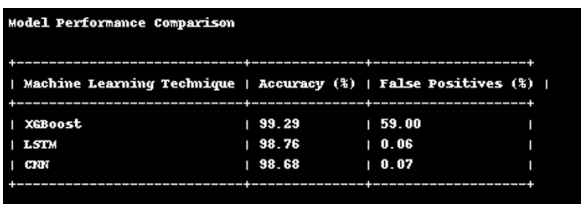


Figure-4 (Confusion Matrix Output for XGBoost)

Then we created an LSTM model with one LSTM layer, two Dense layers (with ReLU activation), and a final Dense layer with softmax activation for multi-class classification. Creates a CNN model with one 1D convolutional layer (Conv1D), a Flatten layer, two Dense layers (with ReLU activation), and a final Dense layer with softmax activation. We then converted the sparse matrices (protocol_final, service_final, flag_final) to dense arrays and combined them with the numerical features (df_xgb_dense) using np.hstack. After that we added the target variable to the DataFrame and saved it as a pickle file. Finally, a comparison of the performances of the models based on their accuracy and false positive rates were printed. The use of PrettyTable enhances the presentation of the results. This was helpful for quickly assessing which model performs better in terms of both overall accuracy and minimizing false alarms (false positives).



Model Performance Comparison

Machine Learning Technique	Accuracy (%)	False Positives (%)
XGBoost	99.29	59.00
LSTM	98.76	0.06
CNN	98.68	0.07

Figure-5 (The Output of the Comparison Report)

For LSTM, while its accuracy (98.76%) is slightly lower than XGBoost, it has an incredibly low false positive rate (0.06%). This makes it very good at identifying real intrusions without raising many false alarms. LSTMs use "gates" (special components within the network) to control the flow of information into and out of the memory cells. This helps them selectively remember or forget information, making them very effective at learning long-term dependencies in data. Like LSTM, CNN achieves high accuracy (98.68%) and a very low false positive rate (0.07%). It might be particularly good at detecting spatial patterns within network traffic. CNNs are exceptional at automatically learning relevant features from raw network traffic data. This eliminates the need for manual feature engineering, which can be time-consuming and requires domain expertise.

IV. SCOPE FOR FUTURE WORKS

This will be the most effective and efficient approach to any network-based systems. There is a need of Multiple high-speed NICs to capture network traffic from various points in our network (e.g., mirrored ports on switches, taps). A multi-core CPU or even a specialized AI accelerator (like a GPU or an FPGA) to handle the computationally intensive machine learning and deep learning algorithms. A signature database which regularly updates database of known attack signatures. A R&D team to integrate the Hardware and Software together in a device which will solely be based as a security hardware product integrated with complex system software.

V. CONCLUSION

The ability of algorithms like XGBoost, LSTM, and CNN to analyze network traffic patterns, identify anomalies, and classify intrusions with high accuracy paves the way for more robust and adaptive security systems. This not only enhances the protection of sensitive data and critical infrastructure but also fosters trust in the digital realm, enabling innovation and growth across various sectors. As these technologies continue to mature and integrate into real-world security solutions, we can expect a paradigm shift in how we defend against cyberattacks. The proactive and adaptive nature of AI-powered security measures will empower individuals and organizations to confidently navigate the digital landscape, driving progress in areas such as finance, healthcare, and communication. In conclusion, the application of machine learning to cybersecurity marks a significant advancement in safeguarding our interconnected world. By embracing these intelligent algorithms, we can create a more secure and resilient technological landscape, ushering in an era where innovation and progress flourish without the constant threat of cyberattacks.

VI. ACKNOWLEDGMENT

I would like to express my sincere gratitude to my guide, Srikanta Sir, for his invaluable support and guidance throughout this research. His thoughtful mentorship, providing me with the space and time needed to explore this topic, has been instrumental in my success. I am deeply grateful to my family for their unwavering love and encouragement. Their constant support during the intense periods of this research provided me with the strength and motivation to persevere. I would also like to extend my thanks to my friend, Arka, for his unwavering belief in me. His friendship and ability to keep me grounded during challenging times were essential to my well-being throughout this journey.

VII. REFERENCES

[1] Cryptography and Network Security (2016) by By S. Bose, P. Vijayakumar

[2] Machine Learning Techniques for Intrusion Detection (2020) by Tameem Ahmad, Mohd Asad Anwar, Misbahul Haque

[3] Network Intrusion Detection using Deep Learning (2018) by by Kwangjo Kim, Muhamad Erza Aminanto, Harry Chandra Tanuwidjaja

[4] A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015 by Atilla Özgür, H. Erdem

[5] Machine Learning Algorithms and Applications (2016) by Anjanna Matta, G. Sucharitha, Mettu Srinivas