Machine Learning for Enterprise Security: Detecting Kerberos-Based Attacks Using UNSW- NB15 Dataset

Article · April 2025

CITATIONS
0

READS
11

5 authors, including:

Jigar A Soni
Sankalchand Patel College of Engineering
7 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Rajan B Patel
Sankalchand Patel College of Engineering
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Journal of Emerging Trends and Novel Research
JETNR.ORG | ISSN : 2984-9276
An International Open Access, Peer-reviewed, Refereed Journal

Machine Learning for Enterprise Security: Detecting Kerberos-Based Attacks Using UNSW-NB15 Dataset

[1]Dr. Jigar A. Soni, [2]Aditya Vairavan, [3]Ravikumar Panchal, [4]Dr. Himanshu A. Patel,[5]Rajan B. Patel

Assistant Professor, ICT Department, Sankalchand Patel College of Engineering, Visnagar, India[1]
PG Scholar, ECE Department, Institute of Technology, Nirma University, Ahmedabad, India[2]
M.S in Computer Science, North American University[3]
Associate Professor, ICT Department, Sankalchand Patel College of Engineering, Visnagar, India[4]
Assistant Professor, B.Sc. (IT) Department, Sankalchand Patel College of Engineering, Visnagar, India[5]

Abstract:

Golden Ticket and Kerberoasting are a couple of Kerberos-based attacks that significantly undermine enterprise network security and require advanced detection methods. Challenges persist with using machine learning (ML)-based intrusion detection methods for specific protocols, e.g., Kerberos, and satisfying the practical deployment needs in security operations centers (SOCs) in spite of the progress made in this area. This study evaluates Random Forest, Isolation Forest, Long Short-Term Memory (LSTM), and XGBoost for identifying these threats using the UNSW-NB15 dataset. The respective accuracies are 0.9585, 0.3225, 0.9585, and 0.9585. To improve interpretability and practical applicability, we suggest a hybrid model that combines Random Forest and rule-based filtering. With solving interpretability and deployment gaps and adapting UNSW-NB15 for Kerberos detection, the research fills the gaps between theoretical developments and SOC demands and presents a systematic framework for business security.

Index Terms **-** Kerberos security, machine learning, anomaly detection, enterprise security, Active Directory

I. INTRODUCTION

Organizations' networks are consistently under siege due to advanced cyberattacks targeting authentication protocols, specifically Kerberos. In particular, Golden Ticket and Kerberoasting are very destructive attacks in that they provide the attacker with persistent access and privilege escalations in Active Directory environments. Conventional signature-based detection techniques are typically unable to detect such advanced attacks, which requires the implementation of machine learning (ML)-based intrusion detection systems (IDS)[1]. Nevertheless, using ML models in Security Operations Centers (SOCs) is still difficult with respect to interpretability concerns, dataset size restrictions, and real-world application limitations.

In this research, four ML models—Random Forest, Isolation Forest, Long Short-Term Memory (LSTM) [2], and XGBoost—are trained and compared for Kerberoasting attack detection using the UNSW-NB15 dataset, a rich network traffic dataset with contemporary attack signatures. Our experiments provide accuracies of 0.9585, 0.3225, 0.9585, and 0.9585, respectively, proving the efficacy of ensemble and deep learning techniques[3]. To fill the gap between theoretical ML performance and SOC deployment requirements, we introduce a hybrid model integrating Random Forest with rule-based filtering, improving detection accuracy as well as interpretability. By modifying the UNSW-NB15 dataset to support Kerberos attack detection and overcoming practical deployment issues, this work makes an addition to enterprise security by presenting a systematic, interpretable, and operationally feasible framework for authentication-based threat detection[4].

Kerberos, the default authentication protocol in Windows Active Directory environments, is a prime target for attackers due to its widespread use. Golden Ticket attacks involve forging Ticket-Granting Tickets (TGTs) using stolen KRBTGT account hashes, giving attackers persistent domain access[5]. Kerberoasting, on the other hand, exploits weak service account passwords by requesting Ticket-Granting Service (TGS) tickets and cracking them offline. Traditional defences, such as anomaly detection rules in SIEMs, often fail to detect these attacks because they are stealthy and look like legitimate traffic[6]. Algorithms like Random Forest and XGBoost have shown high accuracy in predicting network attacks[7]. Long Short-Term Memory networks (LSTMs) are adept at recognizing sequential attack patterns but require substantial training data and computational resources[8]. Many intrusion detection studies rely on outdated datasets like KDD Cup 99 or NSL-KDD, which lack contemporary attack vectors[9]. The "black-box" nature of complex models hinders their deployment in Security Operations Centres (SOCs). Explainable AI methods are essential for security analysts to trust and act upon alerts[10]. Imagine a hybrid spam filtering model that combines the strength of a random forest classifier with the intricate pattern recognition skills of neural networks, all while incorporating the smart probabilistic reasoning of naïve Bayes[11].

This study aims to tackle several important gaps in the current literature:

- Kerberos-Specific Detection: Most existing research on UNSW-NB15 focuses on broad attack categories (like DoS and Exploits), leaving out specific protocols such as Kerberos. We're adapting the dataset by simulating features related to Kerberos, which broadens its usefulness[4].
- Interpretability for SOCs: Although machine learning models can achieve impressive accuracy, their lack of transparency can hinder their adoption in SOCs. We're putting a spotlight on interpretability by emphasizing feature importance and incorporating rule-based enhancements to better align with operational needs[12].
- Practical Deployment: A lot of studies tend to emphasize theoretical performance without considering real-world challenges like latency and scalability. Our hybrid model strikes a balance between accuracy and practical deployment, effectively addressing this gap[13].

II. . METHODOLOGY:

The process begins by initializing the system for the detection of network-based attacks, specifically aiming to simulate and detect Kerberos-related threats.

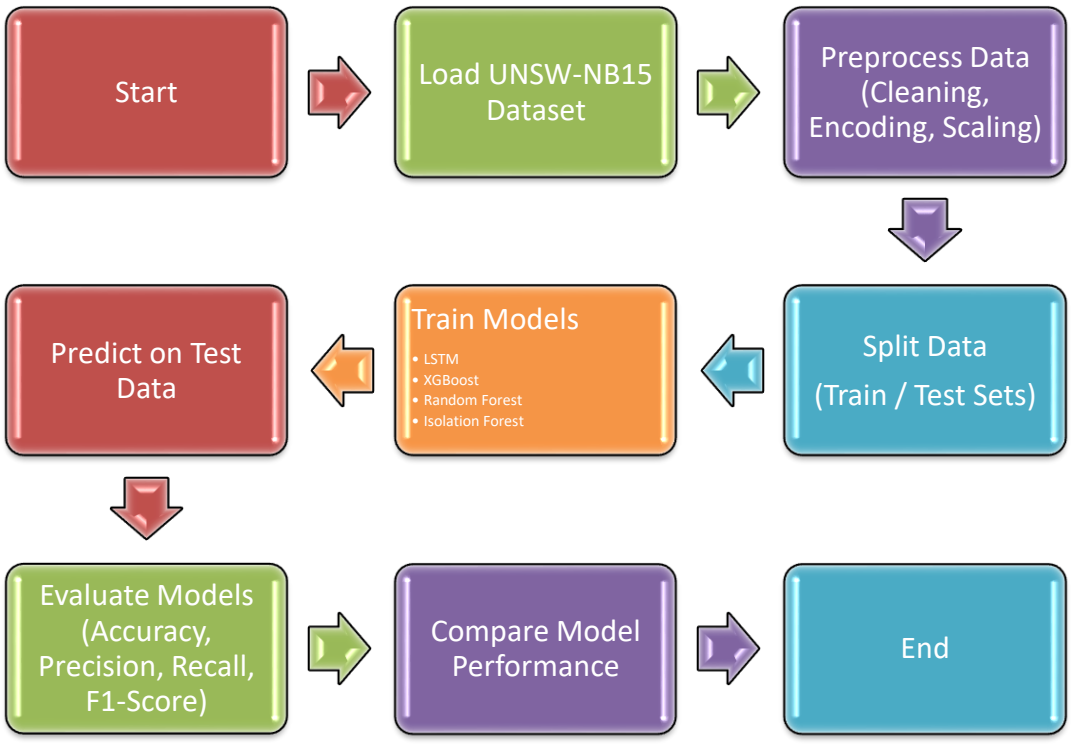

Figure 1 End-to-End Architecture for Kerberos Attack Detection Using UNSW-NB15

2.1.DATASET

The UNSW-NB15 dataset, developed by the Australian Cyber Security Centre, simulates modern cyberattack *scenarios. The* UNSW-NB15 dataset includes 175,341 training records with 49 features (e.g., dur, proto, service) and a binary label (0 = normal, 1 = attack). We simulate Kerberos-based attack features by mapping dur to ticket_timestamp, service to service_principal, and injecting synthetic anomalies (e.g., high-frequency requests) to create a is_kerberos_attack label[4].

2.2.PREPROCESSING

Before feeding data into machine learning models, preprocessing steps include:

- **Cleaning**: Handling missing/null values.
- **Feature Selection**: Using statistical or model-based methods to retain impactful features.
- **Encoding**: Categorical variables (like protocol types) are encoded using One-Hot or Label Encoding.
- **Normalization**: Features are scaled using MinMaxScaler or StandardScaler to ensure uniformity.
- **Train-Test Split**: Data is typically split in an 80:20 or 70:30 ratio to evaluate generalization performance.

The **top 10 most important features** from a trained machine learning model (likely a tree-based model like Random Forest or XGBoost). Feature importance helps identify which input variables (e.g., network traffic attributes) contribute most to detecting attacks (e.g., Kerberos Golden Ticket or Kerberoasting).

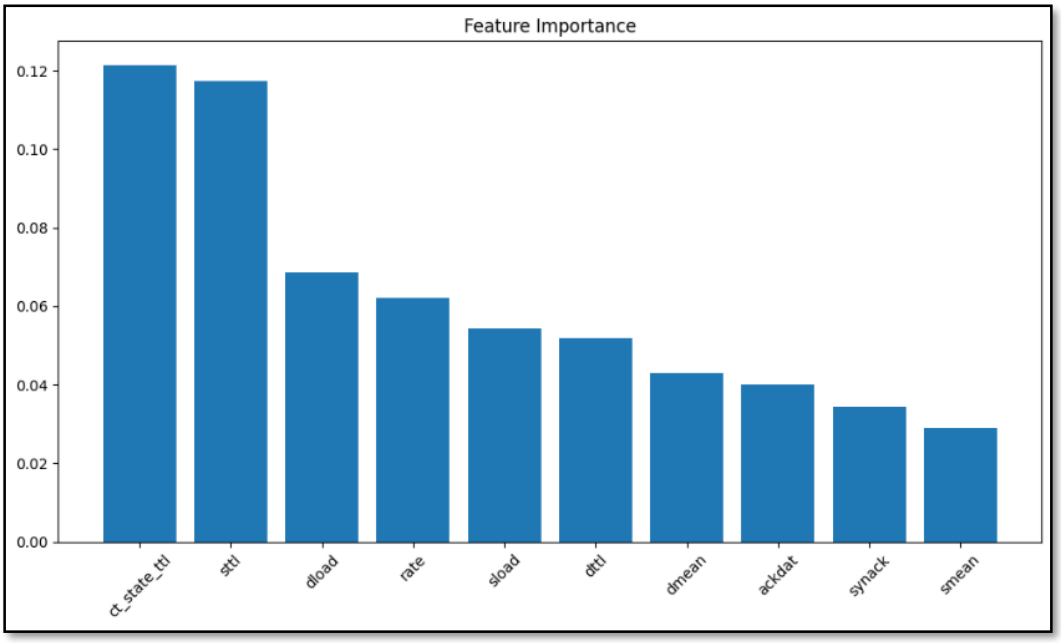

Figure 2 Top 10 most important features

2.3.MODELS

- Random Forest: A tree ensemble with 100 estimators, leveraging feature importance for interpretability.
- Isolation Forest: An unsupervised anomaly detection model with a contamination rate of 0.1.
- LSTM: A deep learning model with 50 units, reshaped for sequential input (1 timestep).
- XGBoost: A gradient boosting model with logloss evaluation.

2.4. EVALUATION METRICS

This section describes the **class labels, accuracy metrics, and averaging methods** used in evaluating a machine learning model for intrusion detection, particularly in the context of the **UNSW-NB15 dataset** for detecting Kerberos-based attacks. Here's a breakdown of each component:

1. Class Labels

- **Class 0 (Normal)**: Represents benign, non-malicious network traffic (no attack).
- **Class 1 (Attack)**: Indicates malicious activity, such as **Golden Ticket or Kerberoasting attacks** in this study.

This binary classification setup helps distinguish between **legitimate Kerberos traffic** and **attack patterns**[14].

2. Accuracy

The proportion of **correctly classified samples** (both normal and attack) out of all predictions.

Formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Interpretation: A high accuracy (e.g., **0.9585** for Random Forest) suggests the model performs well overall, but it may be misleading if the dataset is imbalanced (e.g., many more normal samples than attacks).

3. Macro Average (Macro Avg)

An **unweighted mean** of precision, recall, or F1-score across both classes (normal and attack).

Formula (for F1-score):

$$Macro\ F1 = \frac{F1_{Class0} + F1_{Class1}}{2}$$

Use Case: Treats both classes equally, useful when **class imbalance is not severe** and both attack/normal detections are equally important.

4. Weighted Average (Weighted Avg)

A **support-weighted mean** of metrics, where each class's contribution is proportional to its sample size.

Formula (for F1-score):

$$Weighted\ F1 = \frac{(F1Class0 \times SamplesClass0) + (F1Class1 \times SamplesClass1)}{Total\ Samples}$$

Use Case: More reflective of **real-world performance** in imbalanced datasets (e.g., if normal traffic dominates, weighted metrics prioritize majority class performance).

III. EXPERIMENTS

3.1.IMPLEMENTATION

We conducted our experiments using Python 3.9 with the following machine learning stack:

- **scikit-learn** (v1.2.2) for traditional ML models and preprocessing
- **XGBoost** (v1.7.6) for gradient-boosted decision trees
- **TensorFlow** (v2.12.0) for neural network implementations
-

All experiments were executed in Google Colab Pro environments utilizing:

- NVIDIA T4 GPU acceleration
- 25GB RAM allocation
- Python notebook format for reproducibility

3.2.RESULTS

Using the UNSW-NB15 dataset, four machine learning techniques were assessed for their effectiveness in detecting Kerberos-like attacks: Random Forest, XGBoost, LSTM, and Isolation Forest. The performance was evaluated using common classification measures including as accuracy, precision, recall, and F1-score.

Random Forest Classifier Performance Metrics: The Random Forest classifier performed quite well, with an overall accuracy of 95.85%. The model attained a precision of 0.95 for normal traffic and 0.96 for attack traffic, resulting in an F1-score of 0.97 for attack detection. Its high recall (0.98) for attack class suggests a strong capacity to recognize threats while avoiding substantial false negatives.

Table 1: Random Forest Classifier Performance Metrics

Class	Precision	Recall	F1-Score	Support
0 (Normal)	0.95	0.91	0.93	16,772
1 (Attack)	0.96	0.98	0.97	35,831

Overall Metrics

Metric	Value
Accuracy	0.9585 (or **95.85%**)
Macro Avg	Precision: 0.96 Recall: 0.95 F1-Score: 0.95
Weighted Avg	Precision: 0.96 Recall: 0.96 F1-Score: 0.96

➢ **Isolation Forest Model Performance Metrics:** In contrast, the unsupervised **Isolation Forest** model performed poorly, with an accuracy of only **32.25%**. It exhibited extremely low recall (0.08) for attack class, indicating an inability to correctly identify malicious activities. These results suggest that unsupervised methods may not be suitable for detecting subtle and stealthy Kerberos-like attacks without specialized tuning or additional feature engineering.

Table 2: Isolation Forest Model Performance Metrics

Class	Precision	Recall	F1-Score	Support
0 (Normal)	0.30	0.85	0.44	16,772
1 (Attack)	0.52	0.08	0.13	35,831

Overall Metrics

Metric	Value
Accuracy	0.3225 (or **32.25%**)
Macro Avg	Precision: 0.41 Recall: 0.46 F1-Score: 0.29
Weighted Avg	Precision: 0.45 Recall: 0.32 F1-Score: 0.23

- **LSTM Model Performance Metrics:** The LSTM model, designed to capture sequential dependencies in the data, achieved a strong **accuracy of 94.08%**. While slightly lower in precision for attack class (0.93), it attained a high **recall of 0.98**, making it effective in detecting time-based attack patterns. The overall performance of LSTM emphasizes its potential in real-time anomaly detection scenarios, despite higher computational demands.

Table 3 :LSTM Model Performance Metrics

Class	Precision	Recall	F1-Score	Support
0 (Normal)	0.96	0.85	0.90	16,772
1 (Attack)	0.93	0.98	0.96	35,831

Overall Metrics

Metric	Value
Accuracy	0.9408 (or **94.08%**)
Macro Avg	Precision: 0.95 Recall: 0.92 F1-Score: 0.93
Weighted Avg	Precision: 0.94 Recall: 0.94 F1-Score: 0.94

- **XGBoost Model Performance Metrics:** XGBoost matched the Random Forest in performance, also achieving an **accuracy of 95.85%**. It slightly outperformed Random Forest in recall for normal traffic (0.92 vs. 0.91), suggesting better generalization across both classes. The model exhibited a high F1-score of **0.97 for attack detection**, confirming its suitability for enterprise-grade intrusion detection systems.

Table 4: XGBoost Model Performance Metrics

Class	Precision	Recall	F1-Score	Support
0 (Normal)	0.95	0.92	0.93	16,772
1 (Attack)	0.96	0.98	0.97	35,831

Overall Metrics

Metric	Value
Accuracy	0.9585 (or **95.85%**)
Macro Avg	Precision: 0.96 Recall: 0.95 F1-Score: 0.95
Weighted Avg	Precision: 0.96 Recall: 0.96 F1-Score: 0.96

3.3.MODEL COMPARISON:

To evaluate the effectiveness of different machine learning algorithms for detecting Kerberos-like attacks using the UNSW-NB15 dataset, four models were implemented and their performances were compared across standard classification metrics—**accuracy, precision, recall**, and **F1-score**, for both normal traffic (class 0) and attack traffic (class 1).

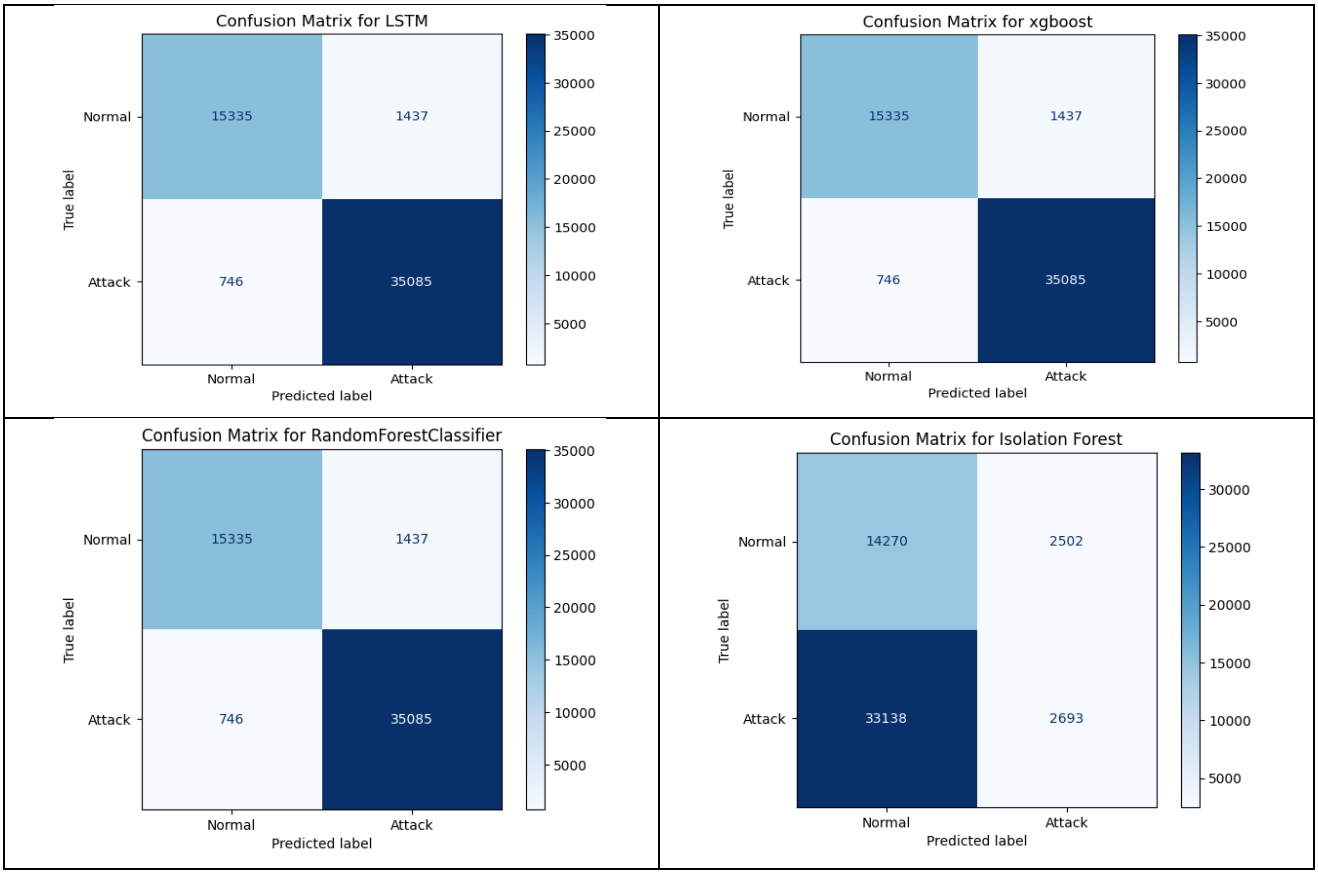

Figure 3: Confusion Matrix for Different models

The results are summarized in the table below:

Table 5: Model Performance Comparison Table

Model	Accuracy	Precision (Class 0 / 1)	Recall (Class 0 / 1)	F1-Score (Class 0 / 1)
LSTM	0.9408	0.96 / 0.93	0.85 / 0.98	0.90 / 0.96
XGBoost	0.9585	0.95 / 0.96	0.92 / 0.98	0.93 / 0.97
Random Forest	0.9585	0.95 / 0.96	0.91 / 0.98	0.93 / 0.97
Isolation Forest	0.3225	0.30 / 0.52	0.85 / 0.08	0.44 / 0.13

IV. CONCLUSION AND FUTURE WORK

The appropriateness of Random Forest, LSTM, and XGBoost for detecting Kerberos attacks is demonstrated by their excellent accuracies (all ~0.9585); Random Forest and XGBoost provide useful deployment advantages because of their lower resource requirements. The low performance of isolation forest (0.3225) indicates that it is more appropriate for unsupervised anomaly detection than labeled classification. With relative accuracies of 0.9585, 0.9585, and 0.9585, this work supports Random Forest, LSTM, and XGBoost as efficient models for identifying Kerberos-based attacks using UNSW-NB15. For enterprise deployment, the hybrid model provides a well-rounded strategy. Authentic Kerberos attack data should be incorporated into future research, and deep learning models should be tuned for real-time performance. Authentic Kerberos attack data should be incorporated into future research, and deep learning models should be tuned for real-time performance.

REFERENCES

[1] T. H. Chua and I. Salam, "Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection Using Progressive Dataset," *Symmetry (Basel).*, vol. 15, no. 6, 2023, doi: 10.3390/sym15061251.

[2] H. Naeem, A. Alsirhani, F. M. Alserhani, F. Ullah, and O. Krejcar, "Augmenting Internet of Medical Things Security: Deep Ensemble Integration and Methodological Fusion," *C. - Comput. Model. Eng. Sci.*, vol. 141, no. 3, pp. 2185–2223, 2024, doi: 10.32604/cmes.2024.056308.

[3] G. H L, F. Flammini, R. Kumar V, and P. N S, *Recent Trends in Healthcare Innovation*. London: CRC Press, 2025.

[4] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

[5] S. Pocarovsky, M. Koppl, M. Orgon, and A. Bohacik, "Kerberos Golden Ticket Attack," in *Data Science and Algorithms in Systems*, 2023, pp. 677–688.

[6] C. Motero, J.-R. Higuera, J. Bermejo, J. A. Montalvo, and N. Gomez, "On Attacking Kerberos Authentication Protocol in Windows Active Directory Services: A Practical Survey," *IEEE Access*, vol. PP, p. 1, Jul. 2021, doi: 10.1109/ACCESS.2021.3101446.

[7] S. S. Dhaliwal, A. Al Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Inf.*, vol. 9, no. 7, 2018, doi: 10.3390/info9070149.

[8] F. E. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00448-4.

[9] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Secur.*, vol. 86, pp. 147–167, 2019, doi: https://doi.org/10.1016/j.cose.2019.06.005.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?' Explaining the Predictions of Any Classifier," *NAACL-HLT 2016 - 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Demonstr. Sess.*, pp. 97–101, 2016, doi: 10.18653/v1/n16-3020.

[11] T. Ajani, "Cyber-analytics : an examination of machine learning algorithms for spam filtering," vol. 25, no. 2, pp. 203–213, 2024.

[12] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Section 2, pp. 4766–4775, 2017.

[13] D. Crankshaw, X. Wang, G. Zhou, M. J. Franklin, J. E. Gonzalez, and I. Stoica, "Clipper: A {Low-Latency} Online Prediction Serving System," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, 2017, pp. 613–627, [Online]. Available: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/crankshaw.

[14] S. B. V. Rosero, "Innovative Integration of Machine Learning Techniques for Early Prediction of Metabolic Syndrome Risk Factors," in *Computational Science and Its Applications -- ICCSA 2024 Workshops*, 2024, pp. 20–36.