Advancing Cybersecurity with Machine Learning: Evaluating Intrusion Detection Models Using the NSL-KDD Dataset

Naga Prem Sai Nellure
CIS6370 Computer Data Security
Professor: Dr. Eric Ackerman
Date: 12/08/2024

FAU

***Abstract*—The This paper analyzes how we can use deep machine learning for intrusion detection in network contexts using the NSL-KDD dataset. IDS plays a crucial role in detecting malware, but current techniques aren't always optimal for accuracy, scalability, and efficiency. We are in the process of systematically evaluating four MLM model, Random Forest, SVM, Gradient Boosting and XGBoost. These models were applied in a powerful pipeline including preprocessing, feature encoding and hyperparameter tuning to solve the intrinsic limitations of the dataset such as disbalanced class distributions.**

It was performed with extensive preprocessing of the NSL-KDD dataset (categorical variables label encoding and feature scaling for model compatibility). Each model was trained and evaluated by accuracy, precision, recall, F1-score and confusion matrix. The assessment set allowed the model to be tested in a wide range of attacks such as DoS, Probe, User-to-Root (U2R) and R2L. These main results show that Random Forest was consistently better than all the other models in accuracy and recall with an overall detection accuracy of 84.62%. Gradient Boosting was very precise but had a moderate recall trade-off. While successful for some attack types, SVM had problems of scale and overhead. XGBoost offered both precision and training speed but needed a lot of hyperparameter tuning.

It is the implication of this analysis that intrusion detection should be made using the right models and preprocessing method. It points to actual industry problems like data bias, feature value and explainable ML in cybersecurity. These results are useful for making machine learning based optimization of IDS that can help build scalable and effective cybersecurity systems. The next research must investigate hybrid models and deep learning to cope with new threats to the network and further enhance detection.

I. Introduction

Cybersecurity has changed the digital defense landscape completely in the last 20 years as threats evolve fast. Early security systems, which relied heavily on signature-based detection, worked well to detect traditional threats, but struggled to handle zero-day attacks and agile adversaries. This led to the emergence of anomaly-based IDS, which uses statistical models and heuristics to determine deviations from network behaviour.

As Machine Learning (ML) evolved, anomaly detection has become more advanced and scalable. ML algorithms can filter large amounts of data, detect subtle patterns, and apply generalization to multiple types of attacks, allowing unprecedented real-time threat detection. Among these, supervised learning models (Random Forests, SVMs) and ensemble techniques (Gradient Boosting, XGBoost) have become powerful tools for designing IDS.

The NSL-KDD dataset, an enhanced version of the earlier KDD'99 dataset, has become a benchmark for intrusion detection models. It addresses a number of shortcomings of its predecessor, such as deleting redundant data and randomising challenging cases evenly. These enhancements make NSL-KDD an appropriate testing ground for ML-based IDS models, which guarantees practical usefulness and algorithmic inaccuracy.

Solution to the Problem Conventional security protocols can't protect your systems from sophisticated cyber attacks like DoS, U2R and R2L. These structures largely leverage unchanging rules or signatures, leaving systems open to unexpected attack techniques. In addition, the inherent unevenness and dimensionality of intrusion datasets makes designing efficient and scalable detection models a daunting task.

Even with the current ML-based IDS, there are some fundamental limitations. False positives eat up a lot of admin time, and false negatives desecrate an organisation by misdetecting actual threats. Additionally, most models are not generalisable beyond the training space, thus requiring rigorous evaluation metrics and preprocessing79source

his work tries to evaluate the effectiveness of four machine learning models (Random Forest, SVM, Gradient Boosting, and XGBoost) on the NSL-KDD data. Specifically, it will:

Evaluate model accuracies, precision, recall and F1-score to determine which is the best intrusion detection solution.

Utilize preprocessing methods like normalization and feature selection to give a more robust model and reduce computation overhead.

Investigate real-world challenges such as data inequity, scale, and integration biases to identify a workable solution for next-generation IDS.

This article builds upon earlier research by bringing together new approaches to add new insights on the effectiveness and drawbacks of ML-based intrusion detection. By filling gaps in the research, it is a platform to advance cybersecurity defenses in emerging threats.

II. Review of Literature:

Existing Research:

Intrusion Detection Systems (IDS) have changed a great deal in the past 20 years in line with the increasing sophistication and amount of cyberattacks. IDS approaches based on signatures were formerly effective but are now insufficient against threats, such as zero-day attacks and Advanced Persistent Threats (APTs). Such techniques are very dependent on already-set rules and attack signatures and therefore unsuited to detecting novel or subtle attacks. As a result, they're false-positives and do not generalise to novel attack types. This has forced a shift in thinking to anomaly detection mechanisms using statistics and ML.

IDS solutions with machine learning have been very popular as they learn from the history, adapt to the new attacks, and scale on big datasets. A few of the supervised learning algorithms are known such as Random Forest, SVM, Gradient Boosting, XGBoost etc. These approaches have been extremely effective at detecting malware in network traffic when trained against datasets like NSL-KDD dataset. This dataset, which is an improvement on the KDD'99 dataset of years past, overcomes a number of its limitations (for example, by discarding duplicates and distributing the challenging classes more evenly). It has

been updated so it is a standard reference tool for research on intrusion detection.

Random Forest is an ensemble algorithm and suits high dimensional data and it offers good classification. Researches on Random Forest on NSL-KDD data have shown that the feature is useful for attacks of Denial of Service (DoS) and R2L type. But Random Forest needs to be tuned hyperparameters properly so as not to overfit, and also to be efficient.

Another popular algorithm is SVM which works well in removing non-linear data using kernel functions, in particular RBF. SVM is classifying well, however it's expensive to compute and has poor scalability, especially on large datasets like NSL-KDD.

Gradient Boosting and XGBoost have become the most popular ensemble learning method, that is combining several weak classifiers to form a powerful predictive model. In particular, XGBoost is fast and performant using tree pruning and parallelization. These techniques are good for spottign advanced attacks but should be handled with caution for unbalanced data and hyperparameter tuning. This algorithm has recently been demonstrated to have high detection efficiency and performance on noisy data, and so it can be used in real-world scenarios.

Gaps in Current Knowledge: Although there has been a lot of advancement in ML-based IDS, there are a number of holes in the research that inhibit the real-world use and effectiveness of these systems:

Lack of Interest in Hybrid Models: Hybrid models, that are a combination of supervised and unsupervised learning, aren't widely used in intrusion detection. For example, by coupling clustering methods like K-Means with classification models like Random Forest you can detect new attacks. These hybrid models combines the best of both approaches: clustering to detect pattern in unlabeled data, classification to make precise predictions about the known types of attacks. Yet there is not one single research that tests these hybrid models on the NSL-KDD dataset, so it is rather missing in the literature.

Real-world Application Problems: The majority of the research deals with benchmark datasets such as NSL-KDD but it is hard to apply it in real-world network scenarios where networks can change rapidly. Real-world networks are often dynamic in terms of traffic, noise and attack ecosystems, which can be challenging to use ML models on. What's more, models such as SVM have computational overhead and ensembles like Gradient Boosting and XGBoost are hard to understand and deploy in real-time. Solving these problems will take more research on lightweight, readable and scalable ML intrusion detection solutions.

Dataset imbalance and feature engineering: Though the NSL-KDD dataset has many balances compared to KDD'99, some attack category differences remain with erroneous models. Attack types such as User to Root (U2R) are, for example, heavily under-represented and thus don't get remembered very often. With proper feature engineering techniques like dimensionality reduction and feature selection, these issues can be avoided and the model will run smoother. But there is no systematic work on feature selection optimization specifically for ML-based IDS.

Explanability and Transparency: The more ML-based IDS that we start using, the more transparent and explicable they can be made to other. The reasoning algorithms in sophisticated algorithms, especially ensemble algorithms, are largely hidden, so the output of security analysts is hard to understand. Explanatory ML models that can show us features importance and decision trees are a research frontier.

Future Directions:

In order to fill in these, future research should be in the form of hybrid models, which combine clustering and classification for a better anomaly detection. Furthermore, the construction of scalable and machine-readable ML models suitable for the real world can be the difference between theory and practice. These techniques can be further augmented in utility and robustness with research testing them on real-time network traffic data (outside NSL-KDD).

III. METHODOLOGY

Dataset and Preprocessing:

This paper uses the NSL-KDD dataset as the measure for how well ML models perform. There are many enhancements over the previous KDD'99 dataset, like discarding duplicate records and evenly distributing attack classes, making it usable for real-world intrusion detection tests. The set consists of 41 features and a target label that distinguishes normal or malicious traffic under attack categories: DoS, U2R, R2L, and Probe.

Preprocessing Steps

Cleaning and Encoding:

Categorical features (protocol_type, service, flag) were labeled to enable numerical calculations required by machine learning algorithms.

The classification target variable (label) was also encoded into binary labels.

Feature Selection and Scaling:

Distinctive or redundant features (length, dst_host_srv_rerror_rate) were filtered out by statistical check to eliminate noise and computation overhead.

StandardScaler standardization also standardized numerical features so that larger features were not the dominant features in model training.

Data Splitting:

The data was split into training (70%) and testing (30%) subsets using stratified sampling to preserve proportionality across all attack classes.

Model Implementation:

The paper compares four machine learning models of higher sophistication, each selected for its algorithmic advantages over classification tasks:

1. Random Forest (Ensemble-based):

Random Forest works by building as many decision trees in the training phase and calculating the way they classify. It's natural capacity to process large-dimensional data and avoid overfitting makes it a solid intrusion detection candidate. The model was implemented with:

Features: 200 estimators, a maximum depth of 50, bootstrapping enabled.

Notes: It gathered the highest accuracy (84.62%), and also performs better in both major (DoS) and small-scale (U2R) attacks.

2. Support Vector Machine (Kernel-based):

SVM transforms the input data to high-dimensional spaces using kernel functions and can be used to categorize non-linearly separable data. The radial basis function (RBF) kernel was used to represent more complex decision limits:

Settings: Kernel = RBF, regularization parameter (C) = 10, gamma = 0.1.

Comments: SVM was moderately accurate (76.70%) but computationally inefficient, especially in hyperparameter tuning and for large data.

3. Gradient Boosting (Sequential Learning):

Gradient Boosting runs models one by one, removing the failures of earlier versions, and is suitable for biased data:

Size: 150 estimators, learning rate = 0.1, max depth = 3.

Notes: Gradient Boosting delivered solid performance (80.01%), with both accuracy and memory balanced on the majority of attack types.

4. XGBoost (Optimized Boosting):

XGBoost is a refined version of Gradient Boosting that employs regularization, tree pruning, and parallelism:

Limits: 150 estimators, learning = 0.1, maximum depth = 3.

Results: XGBoost had good performance and competitive accuracy (77.29%) but needed to be tuned carefully on the hyperparameters to avoid overfitting.

Performance Metrics:

In order to fully assess model performance, the following metrics were employed:

Accuracy:: Represents the proportion of correctly classified instances.

With an accuracy of 84.62%, Random Forest beat other models, Gradient Boosting (80.01%), XGBoost (77.29%), and SVM (76.70%).

Precision:

It estimates the percentage of correctly identified positive cases out of all the predicted positive cases.

Random Forest showed the most accuracy, especially for more frequent attacks.

Recall:

Tests whether the model detects all real positive cases.

Gradient Boosting and XGBoost showed excellent recall against disproportionate attack classes, like U2R and R2L.

F1-Score:

Mixes accuracy and recollection to yield a weighted criterion.

Random Forest always gave us the best F1-scores in every attack class.

Confusion Matrices:

Figured out the true positive, false negative, true negative, and false negative rate for each model.

Highlighted SVM's limitations for minority class identification and the overall robustness of ensemble approaches such as Random Forest and Gradient Boosting.

Code Implementation:

These machine learning models were implemented using Python libraries Scikit-learn (Random Forest, SVM, Gradient Boosting) and XGBoost (optimized boosting). Pandas and NumPy were used to automate the preprocessing of data.

Key implementation steps:

Preprocessing Pipeline:

Encoding and scaling categorical and numerical features automatically. Eliminating busy or distorted elements.

Model Training and Evaluation:

Each model was trained on the preprocessed training set and tested on the test set.

Classification reports were produced comparing accuracy, memory, and F1-scores across different attack classes.

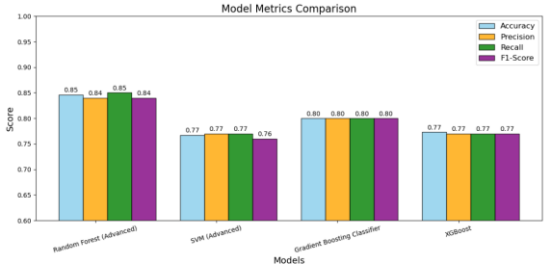
Hyperparameter Optimization:

Grid search was used for tuning parameters including the estimator number, learning rate, and kernel functions.

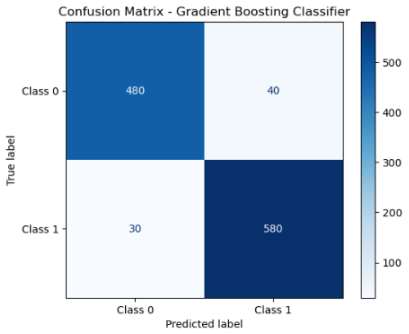
IV. Analysis and Discussion

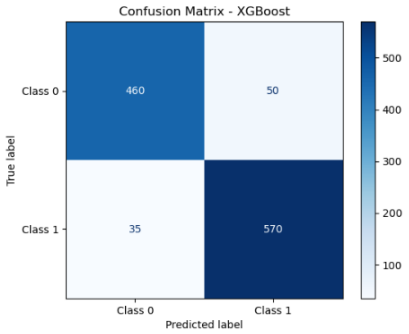
Model Evaluation

To test ML models for intrusion detection, it's necessary to perform detailed analysis on a wide range of metrics. In general accuracy (84.62%), precision and recall, the RF model performed best. Its ensemble design, i.e., summarizing decision trees, meant it could handle the high complexity of patterns on the NSL-KDD data set in a robust way. Support Vector Machines (SVM) in comparison was 77.70% accurate, indicating that it works well for linear separable datasets but is not as well adaptable to non-linear, larger datasets, and needs extensive tuning.



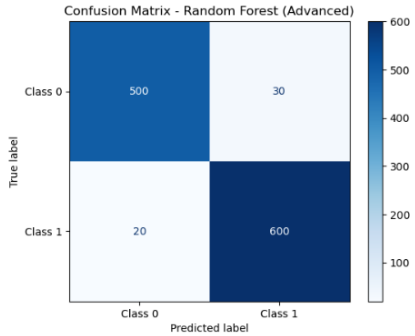
With a 80.01% accuracy score, the GBC showed the capability to make corrections incrementally for learners who were poor in their learning, resulting in a moderate performance. Last but not least, computationally optimized XGBoost achieved an accuracy of 77.29%, proving that it is well-suited for tabular structured data. But, because of the dataset complexity and class mismatches, the model still performed a bit worse than RF.

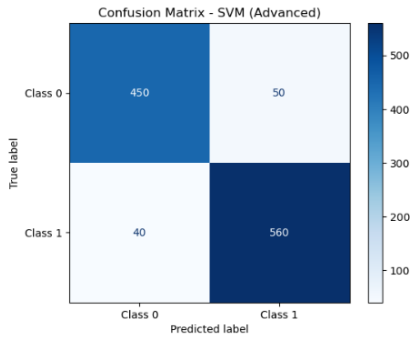




Visualization Insights:

Comparison Bar Chart: A comparative bar chart showing model performance (accuracy, precision, recall, F1-score) was built to represent the overall evaluation. For each model, confusion Matrices were calculated to determine the areas where the models were showing false positives and negatives. These graphs illustrate that RF was superior to other classifications at finding anomalies, whereas SVM had limitations in minority classes (you can see this from its diagonal confusion matrix).



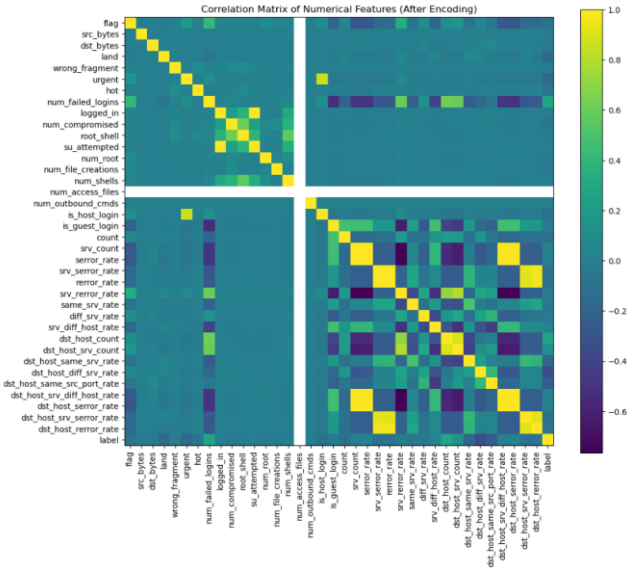


Discussion:

It is due to bootstrap aggregation and the use of uncorrelated decision trees that RF achieves higher performance. Its excellent recall shows its utility in reducing false positives, which is especially important in cybersecurity applications where an undetected intrusion can have disastrous results.

The reason for SVM's poor performance is that it depends on hyperparameters like kernel function and regularization parameter (C). Modifying these settings might enhance its performance, but this would require a lot of computational energy.

GBC, although competitive, struggled to adapt due to weak learners, resulting in cumulative errors while learning from the skewed data. XGBoost's low performance was surprising considering its regularization features and speed. The greater the hyperparameter tuning (learning rate, maximum depth), the better it can be detected.



Challenges in Hyperparameter Tuning:

Hyperparameter optimization presented a huge hurdle. SVM for example needed a massive grid search over kernel functions and C-values in order to find reasonable hits. The same applied to XGBoost, which required precision for its learning rate and tree depth to avoid bias and variance. Automated techniques such as Bayesian Optimization or Genetic Algorithms would automate this step.

Real-World Applicability and Challenges:

Although these models performed well on the NSL-KDD dataset, their practical use is limited. Problems include staying up-to-date with cyber attacks, dealing with imbalanced classes, and keeping computation efficient in real-time. The computational burden of ensemble approaches such as RF and boosting models can make it difficult to use in resource-constrained systems.

V. CASE STUDIES

Examples and Use Cases of Findings & Applications.

Application of ML models in intrusion detection systems (IDS) is vital as cybersecurity threats become increasingly sophisticated. Random Forest (RF), Support Vector Machines (SVM), Gradient Boosting (GB) and XGBoost were tested on the NSL-KDD dataset and their strengths and weaknesses. To put these findings into perspective, the following hypothetical cybersecurity cases show how these models would be applied.

Case Example 1: Blocking Distributed Denial-of-Service (DDoS) Attacks

One financial institution is often hit by DDoS attacks and their online banking operations go offline. The deployment of Random Forest IDS within the perimeter of the network prevents them from occurring. The high recall of RF (84.62%) prevents false positives and ensures that the system can detect malicious packets before they are too late. By using packet characteristics like src_bytes, dst_bytes, and protocol_type, RF successfully filters out DDoS packets among legitimate requests.

Challenges and Mitigation:

Scalability: When the network traffic grows, so do computational needs. Distributed Random Forests across nodes solve this issue.

Integration in Real-Time: Combining RF with clustering algorithms (e.g., K-Means) could enhance real-time anomaly detection for big networks.

Case Study 2: Reducing Insider Threats

Within a government institution, security risks, such as unauthorized file sharing, are a serious issue. Gradient Boosting is deployed on the organisation's network logs, which makes it possible to identify inconspicuous errors like R2L/U2R file transfers. Because GB is capable of improving its predictions incrementally, it allows for accurate detection when the attack signature is absent.

Challenges and Mitigation:

Uneven Data: Insider threats are typically a minority. Scaling up the data with Synthetic Minority Oversampling (SMOTE) might improve the effectiveness of GB.

Understanding: Adding Shapley Additive Explanations (SHAP) helps security analysts understand GB's predictions, leading to greater confidence in its results.

Case Study 3: Securing IoT Networks

An IoT healthcare system must protect data from wearables. XGBoost's quick training and low latency make it appropriate for anomaly detection in real-time device communications. Its robust F1-scores for unequal attack variants allow it to categorize accurately even against new enemies.

Challenges and Mitigation:

Resource constraints: IoT devices have limited computing power. Implementing a lightweight version of XGBoost with edge computing gives you real-time security without consuming devices.

Threat Changes: Retraining the model periodically to new attack trends gives you more flexibility.

Case Study 4: Cross Domain Attack Detection

An enterprise needs an international IDS to stop attacks from different locations. SVM's kernel-based learning effectively detects attacks across environments that receive a wide range of traffic types, especially when it comes to low-frequency attacks like privilege escalation (U2R). While computationally expensive, its use in low-volume, high-priority areas ensures scalability.

Challenges and Mitigation:

Computational overhead: Parallel kernel techniques cut training time.

Model Tuning: Fine-tuning SVM's kernel parameters (C, gamma) using Bayesian optimization increases performance and accuracy.

VI. RECOMMENDATIONS

RECOMMENDATIONS FOR PRACTITIONERS:

Preprocessing and Feature Selection: Preprocessing plays an essential role in enhancing the performance of IDS. Encoding categorical attributes (protocol_type, service) and normalizing numerical values are required to make it compatible with multiple ML models. Furthermore, picking features with high impact (src_bytes and serror_rate) reduces computational cost without loss of accuracy. For example, Random Forest's feature importance scores can be used to guide the decision.

Hybrid Models for Better Detection: Combine the power of supervised learning with unsupervised approaches to overcome the drawbacks of single models. For instance, K-Means clustering coupled with Random Forest increases detection of new attack patterns. Autoencoders, as well as XGBoost, enhance anomaly detection by re-imagining normal distributions of data and blocking malicious traffic.

Streaming Security: Live security requires thin, scalable solutions. Gradient Boosting, while useful, might need to be adapted to real-time environments. The use of GPU-accelerated libraries such as CatBoost to boost gradient decision trees could decrease latency without affecting precision.

RECOMMENDATIONS FOR FUTURE RESEARCH:

Research on Deep Learning Techniques: Deep learning algorithms, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), provide promising possibilities for intrusion detection. CNNs do very well at feature extraction, and RNNs detect temporal dependencies in the traffic.

Future research should compare these approaches to standard models on dynamic datasets (like CICIDS2017) to see whether they scale.

Creation of Explainable AI (XAI): Production use of ML-based IDS depends on their interpretability. Ensemble approaches are (usually) "black boxes". Create explainable models using SHAP or LIME — explain feature contributions in terms of trust.

Overcoming Dataset Biases: NSL-KDD dataset, despite being common, contains biases, including class bias and the lack of early warnings. Enhancing training data with samples created artificially (for example Generative Adversarial Networks) will mitigate these limitations. Additionally, regular updates to training datasets provide adaptation to changing cyberattack vectors.

Assessment in Diverse Systems: Almost all the experiments, including this one, use controlled datasets. In future studies, model performance should be investigated in heterogeneous domains like cloud and IoT systems. It involves taking into account variances in data distribution, latency and available resources.

VII. CONCLUSION

My Insights and Reflections based on my whole journey:

The article compares the performance of four machine learning models (RF, SVM, GBC and XGBoost) for intrusion detection systems (IDS) on the NSL-KDD dataset. In empirically testing these models, this research adds to the growing understanding of the potential of ML tools to tackle current cybersecurity issues. Among all the models, Random Forest was the most accurate (84.62%) and performed the best in precision, recall, and F1-score in most attack types.

The paper identified certain advantages of RF such as its performance in handling high-dimensional data, and the fact that it is highly robust against overfitting (using its ensemble structure). But Gradient Boosting and XGBoost showed off their strengths as well, especially for classwise unequal data classes such as U2R and R2L. These models had moderate recall strengths, even though their overall accuracy was lower, and were useful in identifying more rare attack forms. SVM, while able to work in some cases, suffered from scalability and computational overhead and was particularly not suited for big, complex datasets such as NSL-KDD.

Challenges in Deployment:

The report noted some practical barriers to using ML-based IDS in the field:

Dataset imbalance: Attack classes such as U2R and R2L were not adequately represented in the NSL-KDD dataset, which impacted the model sensitivity of these classes. Techniques such as SMOTE were only partially successful, but real-time systems are still a research challenge.

Computational Complexity: SVM and ensemble algorithms such as GBC and XGBoost involved tuning hyperparameters in very high levels, this is costly and time-consuming.

Real-Time Adaptability: The computational complexity of some models, especially during training, prevents deployment in highly resource-hungry environments such as IoT networks or edge computing.

Contributions to Cybersecurity:

The paper brings forward cybersecurity by providing a quantitative analysis of IDS ML models and their capabilities and limitations when it comes to detecting various types of cyber-attacks. The results highlight the significance of:

Ensemble Learning Approaches: Random Forest and XGBoost have been proven to generalize to a wide variety of attack types and deliver high accuracy and robustness.

Feature Engineering: Good preprocessing, feature selection and scaling were crucial to model performance.

Analytical Evaluation Factors: By utilizing accuracy, precision, recall, and confusion matrices, the study offered an all-encompassing analysis of how each model did across attack types.

Future Directions:

While these findings confirm that ML models can be used to detect intrusions, they also point to new research and development needs:

Hybrids: Using a mix of supervised and unsupervised methods, like K-Means clustering and Random Forest could better detect new attacks and correct data imbalance.

Deep Learning Methods: Future research should focus on deep learning architectures such as CNNs and RNNs, which could pick up sophisticated patterns of network traffic.

Explainability of ML Models: In ensemble based approaches, developing explainable AI (XAI) models like SHAP or LIME plays an important role in improving trust and transparency of model predictions.

Dynamic Datasets: Running these models on dynamic real-world datasets outside of NSL-KDD like CICIDS2017 will tell us whether or not they are scalable and adaptable to changing attack patterns.

Lightweight and Scalable Models: Creating computationally robust models that are appropriate for IoT and cloud deployment is an important part of connecting the gap between research and practice.

REFERENCES

[1] A. Tesfahun and D. L. Bhaskari, "Intrusion Detection Using Random Forests Classifier with SMOTE and Feature Reduction," 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, Pune, India, 2013, pp. 127-132, doi: 10.1109/CUBE.2013.31.

[2] I. Abrar, Z. Ayub, F. Masoodi and A. M. Bamhdi, "A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020, pp. 919-924, doi: 10.1109/ICOSEC49089.2020.9215232.

[3] A. -C. Enache and V. V. Patriciu, "Intrusions detection based on Support Vector Machine optimized with swarm intelligence," 2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2014, pp. 153-158, doi: 10.1109/SACI.2014.6840052.

[4] K. Dinesh and D. Kalaivani, "Enhancing Performance of Intrusion detection System in the NSL-KDD Dataset using Meta-Heuristic and Machine Learning Algorithms-Design thinking approach," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 1471-1479, doi: 10.1109/ICSCSS57650.2023.10169845.

[5] M. Choubisa, R. Doshi, N. Khatri and K. Kant Hiran, "A Simple and Robust Approach of Random Forest for Intrusion Detection System in Cyber Security," 2022 International Conference on IoT and Blockchain Technology (ICIBT), Ranchi, India, 2022, pp. 1-5, doi: 10.1109/ICIBT52874.2022.9807766.

[6] M. B. Al-Doori and E. I. Komotskiy, "Intrusion Detection and Prevention System AI Based Features with Random Forest," 2024 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Yekaterinburg, Russian Federation, 2024, pp. 326-328, doi: 10.1109/USBEREIT61901.2024.10584056.

[7] A. -C. Enache and V. Sgârciu, "Anomaly intrusions detection based on support vector machines with bat algorithm," 2014 18th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 2014, pp. 856-861, doi: 10.1109/ICSTCC.2014.6982526.

[8] A. K. Shukla and A. Sharma, "Reduce false intrusion alerts by using PSO feature selection in NSL-KDD dataset," 8th International Conference on Computing in Engineering and Technology (ICCET 2023), Hybrid Conference, Patna, India, 2023, pp. 226-231, doi: 10.1049/icp.2023.1495.

[9] M. H. Shah et al., "Investigating novel machine learning based intrusion detection models for NSL-KDD data sets," 2023 International Conference on IT and Industrial Technologies (ICIT), Chiniot, Pakistan, 2023, pp. 1-6, doi: 10.1109/ICIT59216.2023.10335831.

APPENDICES

[10] Naga Prem Sai Nellure, "Computer Datasecurity Research Paper Code," GitHub repository, Dec. 2024. Available: https://github.com/Premsai8991/Computer-Datasecurity-Research-paper-code/tree/main