Securing Networks: An In-Depth Analysis of Intrusion Detection using Machine Learning and Model Explanations

Hoang-Tu Vo, Nhon Nguyen Thien, Kheo Chau Mui, Phuc Pham Tien
Information Technology Department
FPT University, Cantho city, Vietnam

***Abstract*—As cyber threats continue to evolve in complexity, the need for robust intrusion detection systems (IDS) becomes increasingly critical. Machine learning (ML) models have demonstrated their effectiveness in detecting anomalies and potential intrusions. In this article, we delve into the world of intrusion detection by exploring the application of four distinct ML models: XGBoost, Decision Trees, Random Forests, and Bagging. And leveraging the interpretability tools LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) to explain the classification results. Our exploration begins with an in-depth analysis of each machine learning model, shedding light on their strengths, weaknesses, and suitability for intrusion detection. However, machine learning models often operate as "black boxes" making it crucial to explain their inner workings. This article introduces LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) as indispensable tools for model interpretability. Throughout the article, we demonstrate the practical application of LIME and SHAP to explain and interpret the output of our intrusion detection models. By doing so, we gain valuable insights into the decision-making process of these models, enhancing our ability to identify and respond to potential threats effectively.**

***Keywords*—Intrusion detection systems; machine learning models; model interpretability; cybersecurity; LIME; SHAP; explainable machine learning models**

I. Introduction

In today's modern economy, the significance of cybersecurity cannot be ignored [1], [2], [3]. It serves as the backbone of a digitally driven world where businesses, governments, and individuals rely heavily on interconnected systems and networks to function efficiently. Cybersecurity not only safeguards sensitive data but also preserves trust, ensuring the smooth operation of financial transactions, the confidentiality of personal information, and the integrity of critical infrastructure. As technology continues to advance, the dependence on digital platforms grows, making cybersecurity an indispensable facet of our economic landscape. Without it, the very foundation of our modern economy would be vulnerable to an array of cyber threats, underscoring its undeniable importance in preserving the integrity and resilience of our interconnected world.

Intrusion Detection Systems (IDSs) [4] play a pivotal role in safeguarding the integrity and security of modern digital environments [5], [6]. These systems act as vigilant sentinels, constantly monitoring network activities and system behaviors to identify any suspicious or malicious actions. In an era where cyber threats have become increasingly sophisticated and prevalent, the importance of IDSs cannot be overstated. They serve as the first line of defense, providing early warnings and alerts to potential security breaches. By promptly detecting and responding to intrusions, IDSs help organizations mitigate risks, protect sensitive data, and maintain the trust of their customers and stakeholders. In essence, IDSs are the guardians of digital landscapes, contributing significantly to the resilience and security of today's interconnected world.

Applying machine learning models to the development of Intrusion Detection Systems (IDS) marks a significant advancement in cybersecurity. These systems leverage the power of data-driven algorithms to identify patterns and anomalies in network traffic, enabling the detection of potential security breaches with a high degree of accuracy. Machine learning models, such as XGBoost [7], Decision Trees [8], Random Forests [9], and Bagging [10], provide the capability to adapt and learn from evolving threats, making them well-suited for the dynamic nature of cybersecurity. By continuously analyzing vast datasets and recognizing subtle deviations from normal behavior, these models enhance the efficiency and effectiveness of intrusion detection. They empower organizations to proactively respond to threats, fortify their defenses, and safeguard critical assets in an increasingly digital world. The application of machine learning in IDS represents a pivotal shift towards more robust and adaptive security measures, essential in countering the ever-growing sophistication of cyber threats.

Machine learning models often operate like black boxes, providing accurate predictions but leaving users in the dark about the reasoning behind those predictions. This opacity can lead to a level of distrust among users, particularly in critical domains like cybersecurity. In such cases, understanding why a model flags certain events as threats or anomalies becomes crucial. This is where interpretable machine learning models and techniques come into play (often called XAI—Explainable artificial intelligence [11]). They offer a crucial layer of transparency by explaining the factors contributing to a model's decision, helping users comprehend the rationale behind predictions. In the world of cybersecurity, where trust and accountability are essential, the incorporation of interpretable models and explanations not only enhances the confidence in machine learning systems but also empowers security practitioners to make informed decisions and take effective actions against potential threats.

The primary purpose of this article is to shed light on

the pivotal role of machine learning models, particularly XGBoost, Decision Trees, Random Forests, and Bagging, in bolstering Intrusion Detection Systems (IDS). It delves into the application of these diverse models in identifying network anomalies and potential intrusions, emphasizing their unique strengths and attributes. Additionally, the article underscores the importance of model interpretability in the context of intrusion detection. It introduces and demonstrates the practical use of interpretability tools like LIME (Local Interpretable Model-agnostic Explanations) [12] and SHAP (SHapley Additive exPlanations) [13] to unveil the decision-making process within these models. By combining the power of machine learning with model transparency, this article equips cybersecurity practitioners with the knowledge and tools to enhance their intrusion detection capabilities, fostering a safer and more secure digital landscape.

The organization of the paper is as follows: Part II provides an in-depth review of the relevant literature, presenting essential contextual information. Part III outlines the methodology employed for classifying types of cyber attacks, encompassing aspects such as the Dataset, Preparation of Data and Evaluation Metrics for the Model. Part IV elucidates the experimental setup and presents the ultimate outcomes. Finally, Part V concludes the research by summarizing the discoveries and delivering concluding insights.

II. Related Works

The development of machine learning and deep learning models has profoundly transformed numerous fields by enabling unprecedented levels of automation, prediction, and data-driven decision-making, such as in healthcare, self-driving car, and agriculture [14–19]. The continuous advancements in these fields highlight the significant impact of machine learning and deep learning on modern technology and industry.

The application of machine learning models to the development of Intrusion Detection Systems (IDS) has emerged as a thriving field of research, characterized by numerous successes. These models, ranging from ensemble methods like Random Forests and Bagging to gradient boosting algorithms such as XGBoost, have demonstrated their prowess in enhancing network security. Researchers have harnessed the adaptability and predictive capabilities of these models to detect even the most intricate forms of cyber threats. By leveraging the wealth of data generated in today's digital environments, machine learning-based IDS have achieved remarkable accuracy rates while minimizing false positives.

Verma, et al. in this paper [20] explores the application of machine learning classification algorithms to enhance IoT security by addressing Denial of Service (DoS) attacks, conducting a comprehensive study of classifiers, evaluating their performance on various datasets, and proposing statistical methods for assessing classifier performance to advance the development of anomaly-based intrusion detection systems for IoT. In the study [21] conducts a thorough survey of machine learning applications in Intrusion Detection Systems (IDSs), introduces two effective approaches for network attack detection using tree-based ensemble learning and optimized training data selection to enhance detection performance while minimizing operational costs. Ziadoon Kamil Maseer, et al. in the paper [22] conducts a comprehensive review of previous studies on AIDS (Anomaly-based Intrusion Detection Systems) by applying 10 popular supervised and unsupervised ML algorithms to evaluate their performance based on various criteria, including true positive and negative rates, accuracy, precision, recall, and F-Score, with the artificial neural network (ANN), decision tree (DT), naive Bayes (NB) emerging as the most effective in detecting web attacks on a real-world network dataset - CICIDS2017. This research [23] evaluates three machine learning algorithms (Decision Jungle, Random Forest, and Support Vector Machine) for building a Machine Learning-based Network Intrusion Detection System (ML-based NIDS), concluding that Support Vector Machine (SVM) exhibits the highest accuracy, precision, and overall effectiveness in detecting network intrusions on the KDD and CIC-IDS2017 benchmark datasets. Authors in the article [24] introduces a hybrid machine learning approach that combines feature selection and data reduction methods, using feature importance decision tree-based methods and the Local Outlier Factor (LOF) method to achieve high accuracy in detecting network anomalies, particularly in the NSL-KDD dataset, demonstrating superior stability compared to other methods, albeit facing challenges in the UNSW-NB15 dataset. In this paper [25] proposes a taxonomy for Intrusion Detection Systems (IDS) based on deep learning, categorizing IDS literature primarily by data objects and evaluates the performance of three machine learning algorithms (Bayes Net, Random Forest, Neural Network) and two deep learning algorithms (RNN, LSTM) using the KDD cup 99 dataset for accuracy assessment with the WEKA program. In this study [26], Support Vector Machine (SVM) and Naïve Bayes machine learning techniques are employed for intrusion detection using the NSL-KDD dataset, with SVM demonstrating superior performance compared to Naïve Bayes, as measured by accuracy and misclassification rates. In the research [27] explores the detection of anomaly traffic in the NSL-KDD dataset using five machine learning techniques, and it reveals that the Random Forest Classifier achieves the highest accuracy and minimal error rates, surpassing the other classifiers, both with and without dataset normalization.

In addition to the extensive research into traditional machine learning approaches, there has been a significant focus on harnessing the potential of deep learning models in the construction of Intrusion Detection Systems (IDS) [28],[29],[30],[31]. Deep learning, a subset of machine learning, involves the use of artificial neural networks with multiple layers to automatically learn intricate patterns and representations from data. These deep neural networks, such as Recurrent Neural Networks (RNNs) [32] and Long Short-Term Memory (LSTM) [33] networks, have demonstrated remarkable capabilities in capturing complex relationships in network traffic data, making them well-suited for detecting subtle and evolving cyber threats.

The main goal of this article is to highlight the essential role of machine learning models, specifically XGBoost, Decision Trees, Random Forests, and Bagging, in strengthening Intrusion Detection Systems (IDS) used for computer security. It delves into how these diverse models can be used to spot unusual activities on computer networks, which might indicate security threats. Additionally, the article emphasizes the importance of making these models easier to understand for cybersecurity experts. It introduces and demonstrates the

practical use of tools like LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) to clarify how these models make decisions. By improving our understanding of these models, we can enhance computer security and make the digital world a safer place.

III. Methodology

A. Data Set

In our research, we evaluate the effectiveness of our methods using the CICIDS2018 dataset, which was originally curated by the University of New Brunswick for the analysis of Distributed Denial of Service (DDoS) data. This dataset is structured into multiple files, each corresponding to specific dates, and is provided in CSV format. The CICIDS2018 dataset encompasses a total of eighty columns, each representing an entry in the Intrusion Detection System (IDS) logging system employed by the University of New Brunswick. The complete dataset is accessible online [34] and https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv. However, for our study, we specifically focus on two CSV files, namely "02-22-2018.csv" and "02-23-2018.csv," which collectively contain 2,097,150 data streams. The dataset's dimensions are (2097150, 80), making it a substantial resource for our research and analysis. Furthermore, it includes four distinct classes: Benign, Brute Force Web, Brute Force XSS, and SQL Injection, making it a valuable resource for exploring various intrusion detection and cybersecurity-related research questions. Table 1 shows further information about the dataset.

TABLE I. Class Distribution of Dataset

Class	Total
Benign	2096222
Brute Force Web	611
Brute Force XSS	230
SQL Injection	87
	2097150

B. Data Preprocessing

Data preprocessing is a crucial step in preparing a dataset for machine learning and analysis. It involves several important tasks to ensure the data's quality and suitability for modeling. First, we need to remove instances with missing class labels, as these are the target values we aim to predict, and without them, the data becomes unusable for supervised learning. Second, we should eliminate instances with missing information, which includes removing rows or samples that have incomplete or null data points, ensuring that our dataset is consistent and complete. Additionally, we should identify and drop constant columns, where the variation is zero, as these columns do not provide any meaningful information for modeling and can be considered redundant. By performing these preprocessing tasks, we can create a clean and reliable dataset ready for further analysis and machine learning tasks.

C. The Predictive Models and Explanation Methods

This article delves into the field of intrusion detection, examining the practical application of four distinct machine learning models: XGBoost, Decision Trees, Random Forests, and Bagging. Additionally, we harness interpretability tools like LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive Explanations) to elucidate the classification results. Comprehensive Machine Learning Workflow for Training an Intrusion Detection Model is presented in Fig. 1 and Flow chart to classify and explain the model's prediction results is presented in Fig. 2.

D. Performance Evaluation Measures

In the context of Intrusion Detection Systems (IDS), the utilization of evaluation metrics like Precision, Recall, F1-score, and Accuracy plays a crucial role in assessing the effectiveness of these systems. Precision measures the proportion of correctly identified intrusion instances among all the instances classified as intrusions. It is essential in IDS to minimize false positives, as they can lead to unnecessary alerts and resource consumption. Recall, on the other hand, evaluates the system's ability to correctly identify all actual intrusion instances. High Recall ensures that the IDS doesn't miss any real threats. F1-score, which is the harmonic mean of Precision and Recall, provides a balanced assessment, especially when there is an imbalance between intrusion and non-intrusion instances. Lastly, Accuracy measures the overall correctness of the IDS predictions, considering both true positives and true negatives. However, in cases of imbalanced datasets where non-intrusion instances are predominant, Accuracy may not be the sole indicator of system performance. In the context of intrusion detection, these evaluation metrics collectively enable researchers and practitioners to comprehensively evaluate the IDS's ability to accurately identify and respond to security threats while minimizing false alarms and missed detections.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 - Score = \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

In which, TP represents True Positive, TN signifies True Negative, FP represents False Positive, and FN stands for False Negative.

IV. Results and Discussion

A. Environmental Settings

The experimental results were obtained by conducting the experiments on the Kaggle platform. The system used for the experiments had 13GB of RAM and a GPU Tesla P100-PCIE with 16GB of memory.

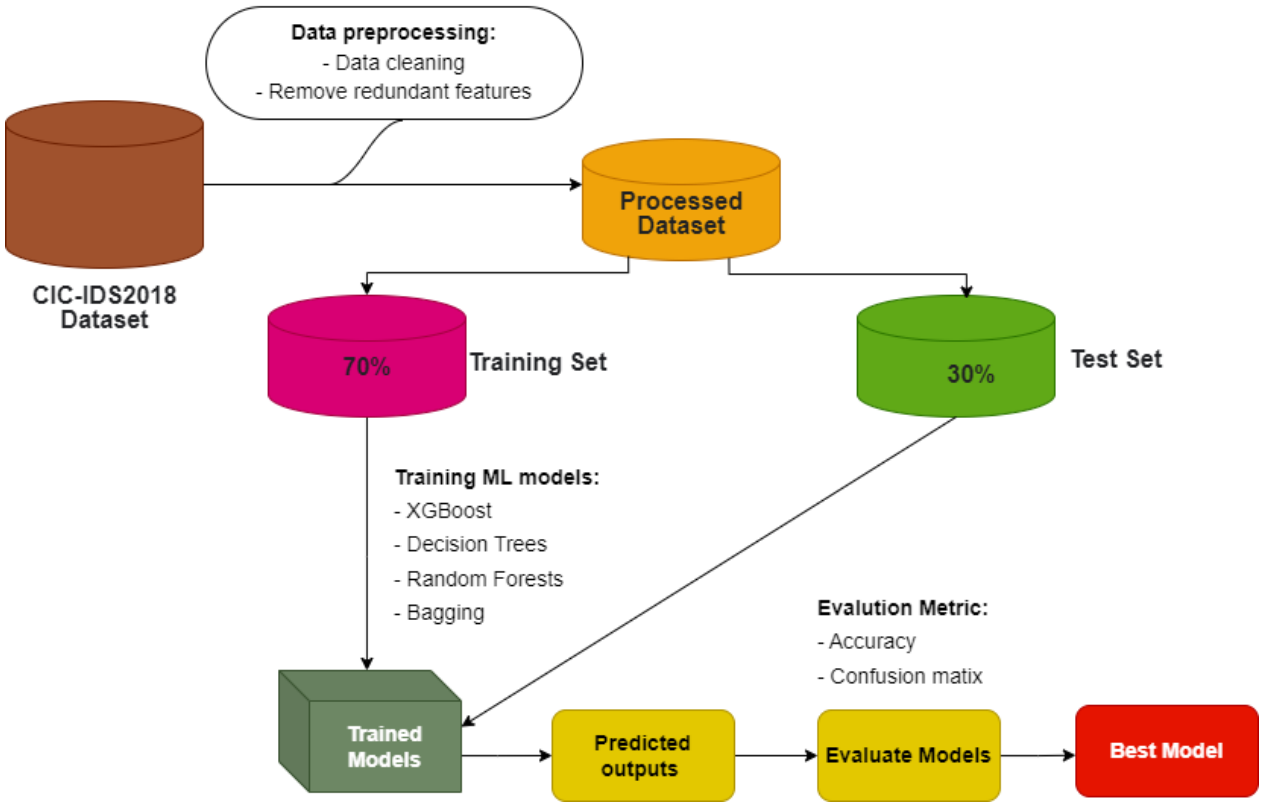

Fig. 1. Comprehensive machine learning workflow for training an intrusion detection model.

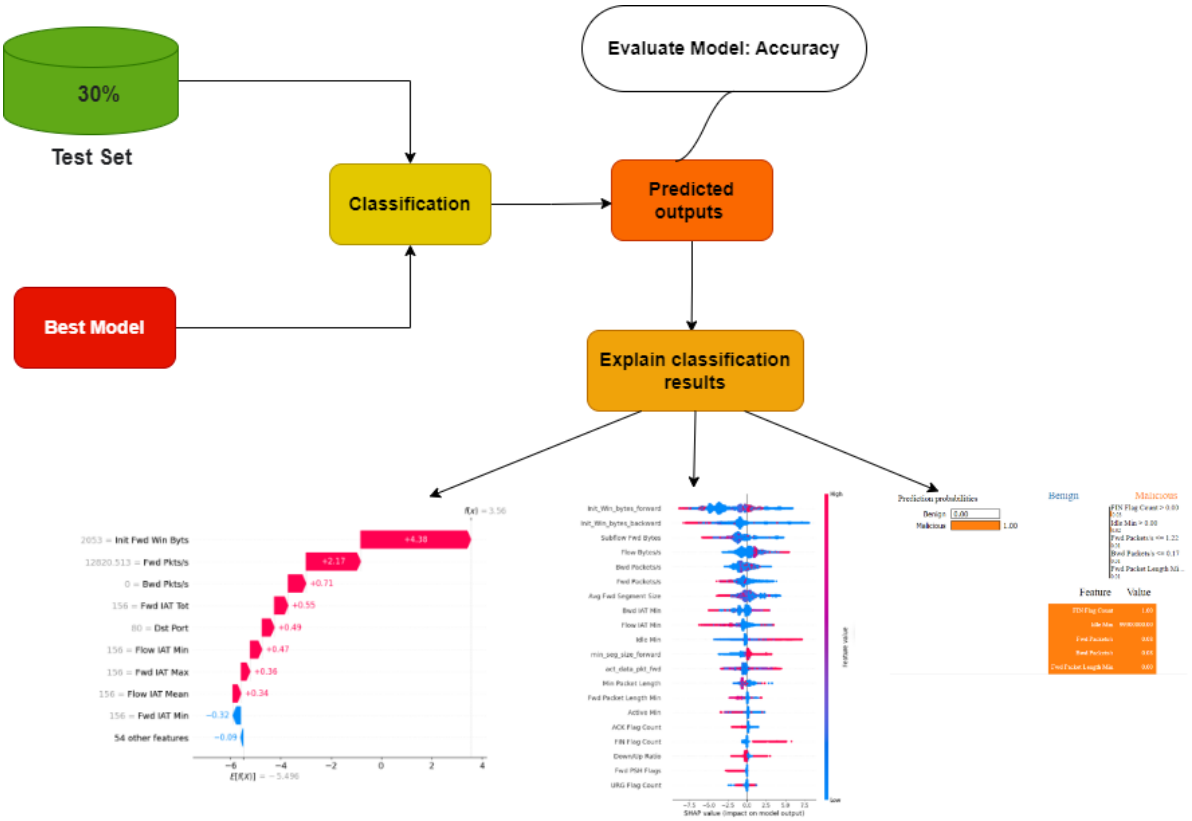

Fig. 2. Flow chart to classify and explain the model's prediction results.

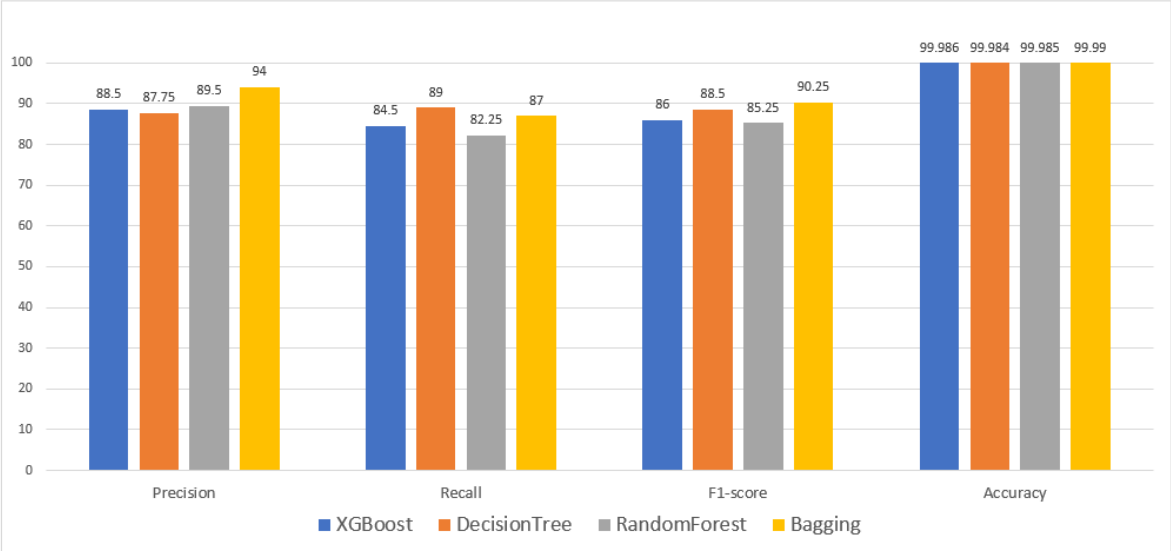

Fig. 3. Comparison chart of precision, recall, F1-score, and accuracy of 4 models.

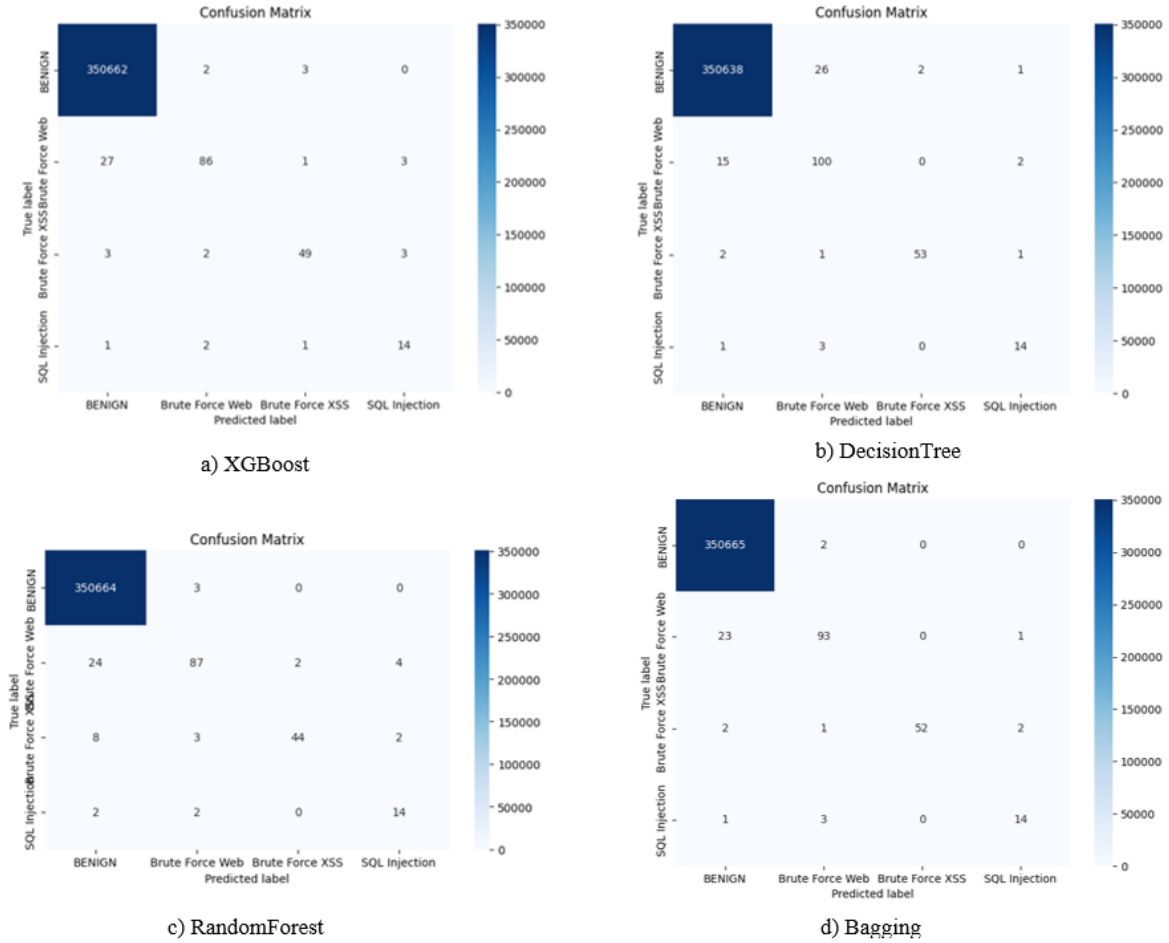

Fig. 4. Confusion matrix of 4 models.

B. Evaluation Overall

In our study, we tried out four different machine learning models – XGBoost, Decision Trees, Random Forests, and Bagging – to tackle the problem of Intrusion Detection. We wanted to see how well each model performs in identifying security threats. After training and evaluating them, we compared their results. This comparison gives us a practical understanding of how effective these models are at spotting intrusions. It helps us see which model might work best for real-world cybersecurity applications, making our research valuable for improving intrusion detection systems. In our performance evaluation of the models, we utilized four key metrics: Precision, Recall, F1-score, and Accuracy, each providing valuable insights into the models' effectiveness for Intrusion Detection. After a thorough analysis, our findings unequivocally demonstrate that Bagging outperforms the other models across all four metrics. Bagging consistently achieved higher Precision, Recall, F1-score, and Accuracy compared to XGBoost, Decision Trees, and Random Forests. These results are visually presented in Fig. 3 and Confusion matrix of four models are presented in Fig. 4.

C. Visualizing the Interpretation of Model Predictions

In this paper, we employ the Bagging model for classification, leveraging its superior performance based on our evaluation criteria, which encompass Precision, Recall, F1-score, and Accuracy. Our choice of the Bagging model stems from its consistent and notable advantage over the other models we considered. Furthermore, we delve into the intricacies of the Bagging model's prediction results using two powerful interpretability techniques: Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP). These interpretability tools provide valuable insights into how the Bagging model makes its predictions, shedding light on the key features and decision factors that drive its classification outcomes. By incorporating LIME and SHAP into our analysis, we aim to enhance our understanding of the model's decision-making process and uncover actionable insights that can inform and strengthen our intrusion detection strategies.

1) LIME: The key idea behind LIME is to approximate the behavior of a complex model using a simpler, more interpretable model locally around a specific instance of interest. By observing how this simplified model behaves in the vicinity of the instance, we gain insights into the factors and features that influence the model's decision for that particular data point.

We utilize network stream index 10782 within our test set, which is designated as 'Brute Force Web'. The classification model consistently predicts this network flow as 'Brute Force Web' with 100% accuracy, relying on the five most critical features: RST Flag Cnt, Dst Port, Bwd IAT Tot, Fwd Pkts/s and Fwd IAT Mean. Detailed results are presented in Fig. 5.

It is evident that the 10782th network flow is confidently predicted as 'Brute Force Web' with a 100% confidence level. This classification decision is based on the following criteria, as validated from the table labeled 'c)': 'RST Flag Cnt' is greater than 0, 'Dst Port' is less than or equal to 80, 'Bwd IAT Tot' is greater than 25202, 'Fwd Pkts/s' is greater than 0.6 and 'Fwd IAT Mean' is greater than 104.

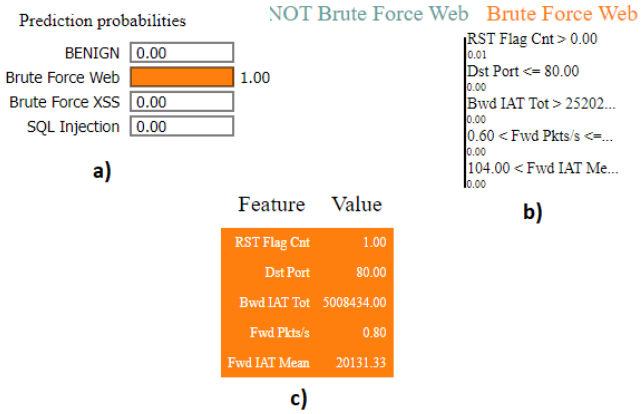

Fig. 5. The outcome comprises three primary elements: a) the model's predictions, b) feature contributions, and c) the actual values for each feature.

Similarly, Network Flow 1735: We use the network stream with index 1735 in the test set labeled 'Brute Force XSS'. The classification model predicts this network flow as a 'Brute Force XSS' network flow with 99% accuracy with the five most important features: RST Flag Cnt, Dst Port, Fwd Pkt Len Mean, Idle Max and Init Fwd Win Byts. Detailed results are presented in Fig. 6.

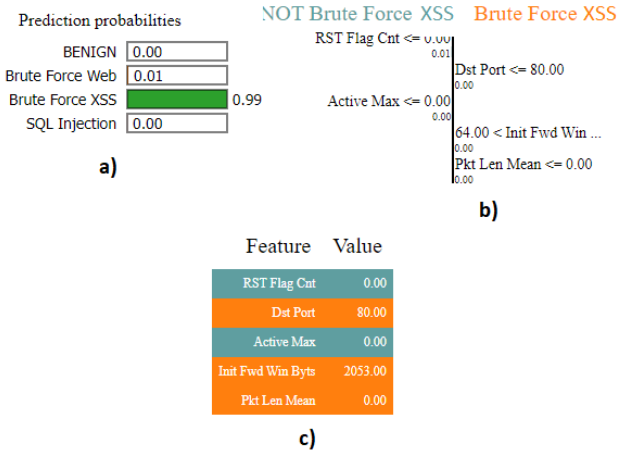

Fig. 6. The outcome comprises three primary elements: a) the model's predictions, b) feature contributions, and c) the actual values for each feature.

2) SHAP: In the context of machine learning, SHAP provides a structured framework to allocate the "credit" or importance of each feature in a model's prediction. It quantifies the contribution of individual features to the model's output, allowing us to understand why a model makes a specific prediction for a given instance. SHAP values allow assessing the significance of each feature in the model's prediction process for each network flow (data point). This helps identify which features strongly influence the prediction outcome, which features have a weak impact, which features counteract the prediction, and which features are not important.

We still use the network stream with index 10782 and use a waterfall chart to explain the prediction results of the

classification model.

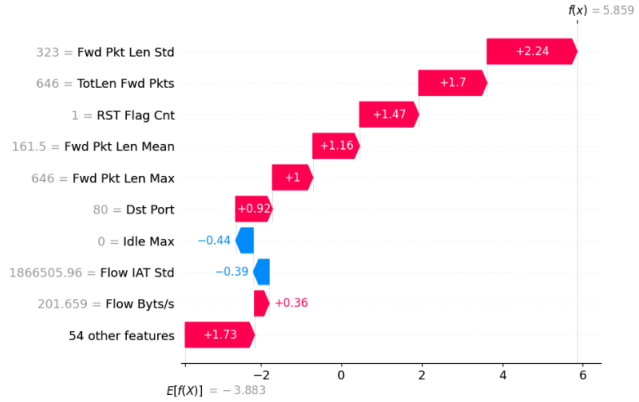

Fig. 7. Waterfall diagram for the 10782nd network flow in the test set.

In Fig. 7, there are 63 Shap values. This chart provides a clear overview of each feature's contribution to the classification model's outcomes. Notably, the feature 'Fwd Pkt Len Std' prominently suggests the possibility of this network flow being classified as 'Brute Force Web.' Following closely in importance are the features 'TotLen Fwd Pkts,' 'RST Flag Cnt,' 'Fwd Pkt Len Mean,' 'Fwd Pkt Len Max,' and 'Dst Port.'

Conversely, the features 'Idle Max' and 'Flow IAT Std' do have some influence in reducing the possibility that this network flow is not 'Brute Force Web,' though their impact is relatively minor.

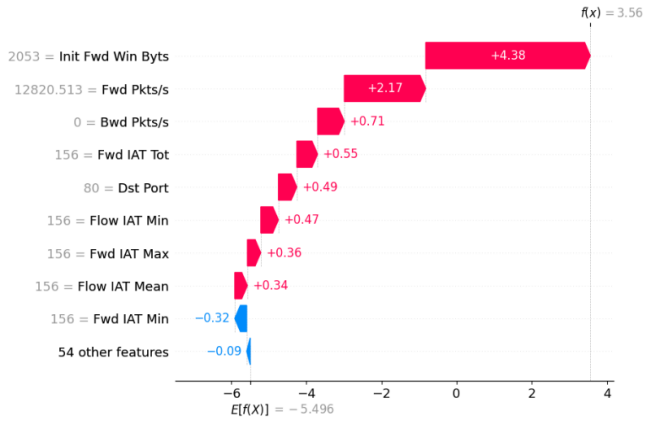

Fig. 8. Waterfall diagram for the 1735nd network flow in the test set.

Likewise, consider Network Flow 1735. Here, we analyze the network stream with the index 1735, sourced from the test set designated as 'Brute Force XSS'. Remarkably, the classification model accurately classifies this network flow as 'Brute Force XSS,' demonstrating an impressive 99% accuracy. Detailed results are presented in Fig. 8.

Evaluate feature importance through Mean SHAP analysis. Within this visualization, features are organized according to their mean SHAP values, with the most critical features positioned at the top and the less influential ones towards the bottom. This representation aids in comprehending the individual feature impacts on the model's predictions. As depicted in Fig. 9, it is evident that the feature 'Idle Std' exhibits substantial positive/negative SHAP values.

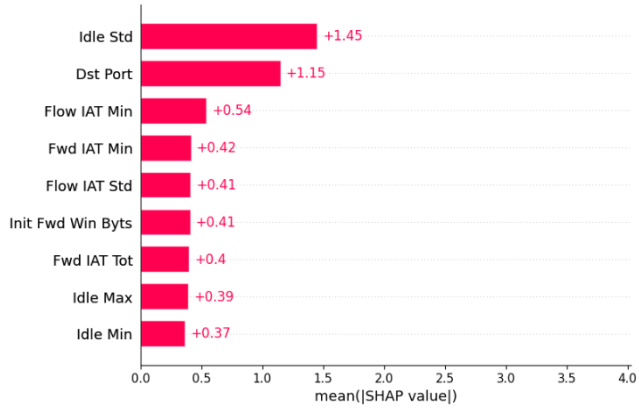

Fig. 9. Average SHAP values showing the most important features.

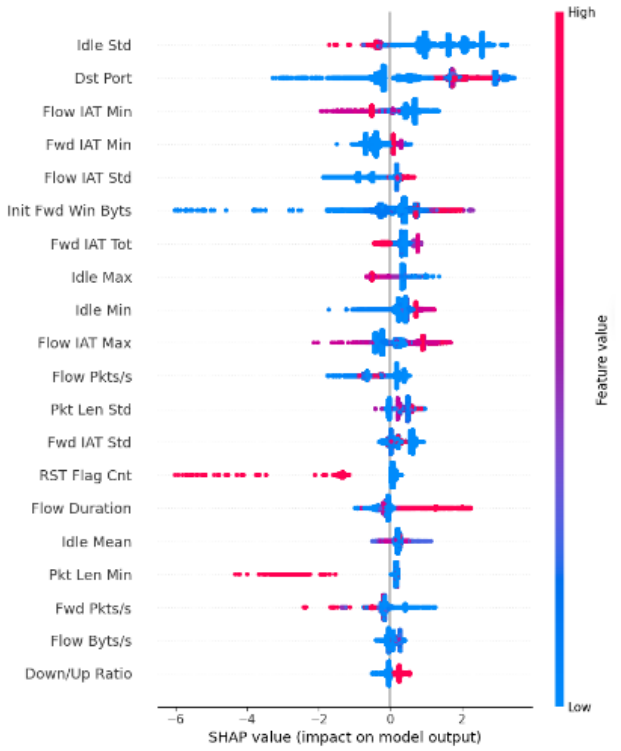

Fig. 10. Average SHAP values showing the most important features.

In Beeswarm plot is presented in Fig. 10, SHAP values show how each feature affects the model's predictions. This plot is great for understanding these relationships. It helps us see how SHAP values connect to the actual feature values, giving us a closer look at each feature's impact on a specific outcome.

In Fig. 10, for example, with the feature 'Idle Std,' as the values of this feature increase (shown in Red), the SHAP value

becomes more negative. Conversely, when the values of this feature decrease (shown in Blue), the SHAP value becomes more positive. This means that higher values of this feature decrease the model's probability of predicting a specific class. Conversely, lower values of this feature increase the model's probability of predicting a specific class.

V. Conclusion

In conclusion, with cyber threats becoming more complex, we urgently need strong Intrusion Detection Systems (IDS). Machine learning (ML) models have proven to be effective in spotting anomalies and potential intrusions.

In this article, we explored four ML models - XGBoost, Decision Trees, Random Forests, and Bagging - and used LIME and SHAP to make sense of their results. We have trained the above models and compared Precision, Recall, F1-score, and Accuracy. Trying to understand how they fit in with intrusion detection.

However, ML models often work like black boxes, so we introduced LIME and SHAP as tools to help us understand how these models make decisions. By applying these tools, we gained valuable insights into the inner workings of our models, giving us an edge in identifying and responding to threats effectively.

The next steps in our journey involve practical implementation and refinement. We will apply the insights gained from our exploration of intrusion detection models and the interpretability tools LIME and SHAP to real-world scenarios. This entails configuring and deploying these models within an operational environment, constantly monitoring their performance, and fine-tuning their parameters to enhance accuracy. Additionally, we will seek to strengthen our models against evolving threats through ongoing research and adaptation, ensuring that they remain effective guardians of digital security.

References

[1] M. Spremić and A. Šimunic, "Cyber security challenges in digital economy," in *Proceedings of the World Congress on Engineering*, vol. 1. International Association of Engineers Hong Kong, China, 2018, pp. 341–346.

[2] I. Vasiu and L. Vasiu, "Cybersecurity as an essential sustainable economic development factor," *European Journal of Sustainable Development*, vol. 7, no. 4, pp. 171–178, 2018.

[3] A. Leahovcenco, "Cybersecurity as a fundamental element of the digital economy." *MEST Journal*, vol. 9, no. 1, 2021.

[4] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[5] J. McHugh, A. Christie, and J. Allen, "Defending yourself: The role of intrusion detection systems," *IEEE software*, vol. 17, no. 5, pp. 42–51, 2000.

[6] S. Thapa and A. Mailewa, "The role of intrusion detection/prevention systems in modern computer networks: A review," in *Conference: Midwest Instruction and Computing Symposium (MICS)*, vol. 53, 2020, pp. 1–14.

[7] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[8] S. B. Kotsiantis, "Decision trees: a recent overview," *Artificial Intelligence Review*, vol. 39, pp. 261–283, 2013.

[9] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[10] ——, "Bagging predictors," *Machine learning*, vol. 24, pp. 123–140, 1996.

[11] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "Xai—explainable artificial intelligence," *Science robotics*, vol. 4, no. 37, p. eaay7120, 2019.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[13] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[14] S. Satpathy, O. Khalaf, D. Kumar Shukla, M. Chowdhary, and S. Algburi, "A collective review of terahertz technology integrated with a newly proposed split learningbased algorithm for healthcare system," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 1–9, 2024.

[15] H.-T. Vo, T. N. Hoang, and L.-D. Quach, "An approach to hyperparameter tuning in transfer learning for driver drowsiness detection based on bayesian optimization and random search," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, 2023.

[16] A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agricultural Technology*, vol. 3, p. 100083, 2023.

[17] H.-T. Vo and L.-D. Quach, "Advanced night time object detection in driver-assistance systems using thermal vision and yolov5," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, 2023.

[18] H.-T. Vo, N. N. Thien, and K. C. Mui, "Tomato disease recognition: Advancing accuracy through xception and bilinear pooling fusion," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 8, 2023.

[19] ——, "A deep transfer learning approach for accurate dragon fruit ripeness classification and visual explanation using grad-cam." *International Journal of Advanced Computer Science & Applications*, vol. 14, no. 11, 2023.

[20] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for iot applications," *Wireless Personal Communications*, vol. 111, pp. 2287–2310, 2020.

[21] Q.-V. Dang, "Studying machine learning techniques for intrusion detection systems," in *Future Data and Security Engineering: 6th International Conference, FDSE 2019, Nha Trang City, Vietnam, November 27–29, 2019, Proceedings 6*. Springer, 2019, pp. 411–426.

[22] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the cicids2017 dataset," *IEEE access*, vol. 9, pp. 22 351–22 370, 2021.

[23] A. H. Azizan, S. A. Mostafa, A. Mustapha, C. F. M. Foozy, M. H. A. Wahab, M. A. Mohammed, and B. A. Khalaf, "A machine learning approach for improving the performance of network intrusion detection systems," *Annals of Emerging Technologies in Computing (AETiC)*, vol. 5, no. 5, pp. 201–208, 2021.

[24] A. A. Megantara and T. Ahmad, "A hybrid machine learning method for increasing the performance of network intrusion detection systems," *Journal of Big Data*, vol. 8, no. 1, pp. 1–19, 2021.

[25] S. V. Amanoul, A. M. Abdulazeez, D. Q. Zeebare, and F. Y. Ahmed, "Intrusion detection systems based on machine learning algorithms," in *2021 IEEE international conference on automatic control & intelligent systems (I2CACIS)*. IEEE, 2021, pp. 282–287.

[26] A. Halimaa and K. Sundarakantham, "Machine learning based intrusion detection system," in *2019 3rd International conference on trends in electronics and informatics (ICOEI)*. IEEE, 2019, pp. 916–920.

[27] F. Yihunie, E. Abdelfattah, and A. Regmi, "Applying machine learning to anomaly-based intrusion detection systems," in *2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. IEEE, 2019, pp. 1–5.

[28] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, pp. 239–247, 2021.

[29] G. C. Fernández and S. Xu, "A case study on using deep learning for network intrusion detection," in *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 1–6.

[30] S. Al-Emadi, A. Al-Mohannadi, and F. Al-Senaid, "Using deep learning techniques for network intrusion detection," in *2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT)*. IEEE, 2020, pp. 171–176.

[31] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," in *2017 IEEE international conference on big data and smart computing (BigComp)*. IEEE, 2017, pp. 313–316.

[32] D. Mandic and J. Chambers, *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. Wiley, 2001.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] Cse-cic-ids2018 on aws, https://www.unb.ca/cic/datasets/ids-2018.html. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.htm