IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 January 2024, accepted 4 March 2024, date of publication 14 March 2024, date of current version 20 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3375878

RESEARCH ARTICLE

A Novel Approach for Real-Time Server-Based Attack Detection Using Meta-Learning

FURQAN RUSTAM[1], ALI RAZA[2], MUHAMMAD QASIM[2], SARATH KUMAR POSA[3], AND ANCA DELIA JURCUT[1], (Member, IEEE)

[1]School of Computer Science, University College Dublin, Belfield, Dublin 4, D04 V1W8 Ireland
[2]Department of Software Engineering, University Of Lahore, Lahore 54000, Pakistan
[3]Department of Information Science, University of Arkansas at Little Rock, Little Rock, AR 72204, USA

Corresponding authors: Anca Delia Jurcut (anca.jurcut@ucd.ie) and Furqan Rustam (furqan.rustam1@gmail.com)

This work was supported by University College Dublin, Ireland.

ABSTRACT Modern networks are crucial for seamless connectivity but face various threats, including disruptive network attacks, which can result in significant financial and reputational risks. To counter these challenges, AI-based techniques are being explored for network protection, requiring high-quality datasets for training. In this study, we present a novel methodology utilizing a Ubuntu Base Server to simulate a virtual network environment for real-time collection of network attack datasets. By employing Kali Linux as the attacker machine and Wireshark for data capture, we compile the Server-based Network Attack (SNA) dataset, showcasing UDP, SYN, and HTTP flood network attacks. Our primary goal is to provide a publicly accessible, server-focused dataset tailored for network attack research. Additionally, we leverage advanced AI methods for real-time detection of network attacks. Our proposed meta-RF-GNB (MRG) model combines Gaussian Naive Bayes and Random Forest techniques for predictions, achieving an impressive accuracy score of 99.99%. We validate the efficiency of MRG using cross-validation, obtaining a notable mean accuracy of 99.94% with a minimal standard deviation of 0.00002. Furthermore, we conducted a statistical t-test to evaluate the significance of MRG compared to other top-performing models.

INDEX TERMS Network security, Wireshark, machine learning, network dataset, intrusion detection.

I. INTRODUCTION

Networking has become an integral part of modern society, enabling seamless communication and interaction between individuals, organizations, and systems. It plays a pivotal role in diverse aspects of life, including personal, professional, and social domains, fostering collaboration, knowledge exchange, and resource-sharing opportunities [1], [2], [3]. However, this pervasive connectivity also exposes networks to a myriad of malicious threats, leading to network attacks that can compromise the confidentiality, integrity, and availability of sensitive information [4].

The growing sophistication of network attacks, such as denial-of-service (DoS), distributed denial-of-service (DDoS), and various intrusion attempts, poses significant challenges for maintaining the security and stability of network infrastructures [4], [5]. These attacks can result in severe consequences, ranging from financial losses and service disruptions to data breaches and reputational damage [6]. According to a report [7], only in June 2023, there were 79 reported security incidents, compromising 14,353,113 records. The total number of data breaches in 2023 reached 607, with 466,078,044 records breached so far. In another report [8], in 2022, the average cost of a data breach rose to a record high of US$4.35 million, with experts predicting potential costs of $5 million in 2023. These statistics emphasize the urgent need for robust security systems to protect network systems from malicious actors and hackers. Securing networks against cyber threats is becoming increasingly urgent to protect sensitive information and prevent financial and reputational damage.

Various types of network attacks target different aspects of network communication and infrastructure. Prominent attack types include file transfer (FTP) attacks [9], [10], [11], [12], remote administration (SSH) attacks [13], web hosting (HTTP) attacks [14], and domain name resolution

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Pau.

(DNS) attacks [15]. These attacks exploit vulnerabilities in protocols like FTP, SSH, and HTTP, as well as target the DNS infrastructure, leading to unauthorized access, data manipulation, and service disruption. Many researchers are working on these kinds of attacks by proposing AI-based approaches to protect network systems such as study [16] utilized machine learning algorithms to detect man-in-the-middle (MTM) and DoS attacks on physically connected devices. The study [17], works on detecting SSH and FTP brute force attacks in large-scale datasets using machine learning techniques. Another study [18], works on a machine learning-based framework to detect DDoS/DoS attacks by combining principal component analysis (PCA) and singular value decomposition (SVD) features.

In comparison to recent studies on these attacks, our research stands out from existing studies as it focuses on recent and real-time server-based network traffic by generating a new dataset for detecting attacks. Unlike previous research that relies on benchmark datasets, our approach utilizes up-to-date data, making it more relevant and effective in identifying and mitigating modern network threats. By employing machine learning techniques on this novel dataset, we aim to provide more accurate and timely detection of attacks, enhancing network security in the face of evolving cyber threats. The primary contribution of our proposed research study is followed as:

- Conducted a thorough comparative analysis of existing network intrusion datasets to identify their limitations and shortcomings.
- Developed a virtual network infrastructure using Ubuntu Base Server to create a real-time network attack dataset. The attacks were initiated from a Kali Linux attacker machine and directed towards the Ubuntu network. Network traffic during each attack was meticulously recorded using the Wireshark tool.
- Generated a novel dataset featuring UDP, SYN, and HTTP flood network attacks based on the captured network attack traffic. The ZUI tool was utilized to convert the pcapng network traffic files into Comma Separated Values (CSV) format for further analysis.
- Employed a machine learning methodology to validate the dataset's patterns and assess its suitability for training machine learning models, ensuring the dataset's effectiveness in model development.
- Introduced an innovative approach in the form of meta-RF-GNB (MRG), a meta-learning-based framework designed for efficient network attack detection. The MRG combines Gaussian Naive Bayes and Random Forest techniques to make accurate predictions, further enhancing the predictions' accuracy by feeding them into a Random Forest-based meta-learner.

The rest of the study is divided into the following sections: Section II provides a comparative analysis of state-of-the-art studies on network attack datasets. Section III-A presents the methodology for generating the novel dataset proposed in this research. Section IV, presents the results of machine learning models on the generated dataset, and Section V presents the research conclusions and outlines future directions.

II. LITERATURE REVIEW

In this section, we conduct analysis to identify trends, methodologies, and advancements in network security with a focus on detecting and mitigating network attacks. We assess various network datasets, evaluate approaches, and perform a comparative literature review of Network Intrusion datasets, comparing them with our generated dataset, as summarized in Table 1.

A. STATE OF THE ART INTRUSION DATASETS

There are many datasets available publically to proposed intrusion detection systems and have their strength and limitations. The KDD'99 dataset [19], widely used for intrusion system evaluation, consists of 41 traffic features categorized into basic, traffic, and content. It contains normal data and four attack types: DoS, R2L, U2R, and probe attacks. However, a major limitation is the high redundancy, with about 78% duplication in the training set and roughly 75% in the testing file. This redundancy negatively impacts detection accuracy, particularly for low-attack categories like R2L and U2R. Similarly, NSL-KDD [20] is an improved version of the KDD'99 dataset, addressing issues like duplicate records. It comprises two subsets: training and testing sets, with the testing set featuring 17 additional attack types not present in the training set. Both KDD'99 and NSL-KDD, despite their wide use in intrusion detection studies, do not accurately represent current network traffic trends due to their age, having been generated over two decades ago.

The Kyoto 2006+ dataset [21] is derived from Kyoto University's honeypot servers, capturing real-time network traffic between November 2006 and August 2009. It features 24 statistical attributes, with 14 overlapping the KDD dataset. However, it suffers from an imbalanced class distribution, primarily comprising malicious data. Furthermore, it lacks specific information about the types of attacks present, limiting its utility for comprehensive intrusion detection evaluation. Despite its real traffic data source, Kyoto 2006+ does not specify the attack types within the dataset. The ISCX2012 dataset [22] was created in a simulated network environment, featuring Alpha profiles for attack traffic and Beta profiles for normal traffic. It primarily focuses on DoS attacks and brute force attacks, including 20 packet features. However, its limitation lies in the limited diversity of DoS attacks, which may not cover vulnerabilities across different OSI layers. The dataset exclusively contains HTTP traffic, potentially not representing modern network traffic patterns. These characteristics align with those observed in the KDD'99 and NSL-KDD datasets.

The UNSW-NB15 dataset [23] is a valuable resource for intrusion detection research. It includes real-world network traffic, encompassing both normal and attack traffic, with various attack scenarios like DoS, DDoS, and exploitation attacks. Researchers and practitioners commonly use it to

TABLE 1. The comparison of existing testbeds with the proposed study for network intrusion detection.

Dataset	Year	No. of Attributes	Labelled	Network Environment	File Format	Covered Attacks	Limitation
KDD'99 [19]	1998	41	Yes	Conventional Network	Others	DoS, Probe, R2L, U2R	High Redundancy, Outdated Data, Limited Diversity of Attacks, Imbalanced Class Distribution, Lack of Real-Time Data, Limited Coverage of Modern Protocols, Limited Features
NSL-KDD [20]	2009	41	Yes	Conventional Network	Others	DoS, Probe, R2L, U2R	Limited Representation of Real-World Traffic, Outdated Data, Limited Diversity of Attacks, Imbalanced Class Distribution, Lack of Encrypted Traffic, Limited Features, Limited Network Environment
Kyoto dataset 2006+ [21]	2006	24	Yes	Conventional Network	Others	Numerous Attacks against honeypots	Imbalanced Class Distribution, Lack of Specific Attack Type Information, Limited Coverage of Modern Network Traffic Patterns (data collected up to 2009)
ISCX2012 [22]	2012	Not known	Yes	Conventional Network	Packet Flow	HTTP, DoS, DDos with IRC botnet, SSH, infiltration	Limited Diversity of DoS Attacks Exclusively Comprises HTTP Traffic, Limited Number of Packet Features
UNSW-NB15 [23]	2015	49	Yes	Conventional Network	Packet Flow & CSV	DoS, DDoS, exploitation attacks.	The dataset does not contain encrypted traffic (e.g., HTTPS) which may limit its representation of modern network traffic patterns and imbalanced Class distribution.
CICIDS 2017 [24]	2017	83	Yes	Conventional Network	Packet Flow	DoS, DDoS, SQL injection, infiltration, portscan	Disparity in the number of features compared to ISCX2012 dataset, Incorporation of the HTTPS Beta profile, Generation of normal traffic behavior based on profile scripts, Presence of numerous redundant records in the dataset.
CSE-CIC-IDS2018 [25]	2018	83	Yes	AWS Platform	Packet Flow	DoS, DDoS, SQL injection, infiltration, portscan	Subject to similar inherent issues as CICIDS 2017, Limited diversity in attack scenarios.

develop and evaluate intrusion detection systems and network security algorithms. However, it lacks encrypted traffic (e.g., HTTPS), potentially limiting its representation of modern network traffic patterns.

CICIDS 2017 [24] broadens attack scenario coverage, but has limitations. It's an extension of ISCX2012, resulting in a notable disparity in feature count, as it includes over 80 flow-based features compared to ISCX2012's 20 packet features. Additionally, the dataset integrates the HTTPS Beta profile to reflect increasing HTTPS use. Normal traffic generation relies on profile scripts, potentially lacking real-world complexity. It also contains numerous redundant records, potentially irrelevant for intrusion detection training.

The CSE-CIC-IDS2018 dataset [25] is a collaborative effort between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC). It shares similarities with the CICIDS 2017 dataset but is implemented on the Amazon Web Services (AWS) computing platform. Profiles were used to systematically create the dataset, including B-profiles for normal traffic generation and M-profiles to simulate attack scenarios. While it encompasses the same attack scenarios as CICIDS 2017, it may also face similar inherent issues encountered in that dataset.

B. MACHINE LEARNING FOR INTRUSION DETECTION

Intrusion detection using machine learning has garnered significant attention from researchers, with many utilizing the aforementioned dataset and even collecting their own data for experimentation. For example, in a study by Rustam and Jurcut [26], they focus on multi-environment malicious traffic detection using ensemble models and particle swarm optimization (PSO). Their experiments involve the use of datasets such as UNSW-NB15, IoTID-20, and an SDN-based dataset. Through the combination of these datasets, they successfully generate novel multi-environment traffic and achieve a remarkable 0.989 accuracy using ensemble models. Similarly, in another study by Sharma et al. [27], a novel intrusion detection system is introduced for IoT networks, employing deep learning techniques, specifically utilizing a filter-based feature selection Deep Neural Network (DNN) model. Their approach, evaluated on the UNSW-NB15 dataset, attains an impressive 84% accuracy in identifying network intrusions.

The study by Kilichev and Kim [28] investigates hyperparameter optimization in one-dimensional convolutional neural networks (1D-CNNs) for network intrusion detection. They utilize genetic algorithm (GA) and particle swarm optimization (PSO) on three prominent datasets, evaluating performance metrics such as accuracy, loss, precision, recall, and F1-score. Their results reveal significant enhancements in all metrics, with GA and PSO achieving notable accuracies of 99.31% and 99.28% on the UNSW-NB15 dataset. Another study [29] focuses on a multi-environment dataset for detecting malicious traffic in networks. This research combines IoT-ID20 and UNSW-NB15 datasets to create a

multi-environment dataset. They employ an extra trees classifier (ETC) along with a novel data balancing technique called Synthetic Data Augment Technique (S-DATE), achieving an accuracy of 0.983 using S-DATE and ETC.

In a similar vein, another study [30] introduces a deep learning-based Intrusion Detection System (DL-IDS). Many existing Intrusion Detection Systems in the literature often lack optimal feature learning and data set management, which can substantially affect attack detection accuracy. The proposed approach combines the spider monkey optimization (SMO) algorithm for feature selection and the stacked-deep polynomial network (SDPN) for classification. SMO identifies optimal features in the datasets, while SDPN categorizes data as either normal or anomalies. The DL-IDS excels in detecting various anomalies, including denial of service (DoS), user-to-root (U2R) attacks, probe attacks, and remote-to-local (R2L) attacks.

1) GAPS AND LIMITATIONS

Throughout our analysis of the benchmarks dataset, we find some common limitations such as High Redundancy, Outdated Data, Limited Diversity of Attacks, Imbalanced Class Distribution, Lack of Real-Time Data, Limited Coverage of Modern Protocols, Limited Features. To address these limitations, this study focuses on creating a novel dataset based on real-world network scenarios and recent attack patterns. Our generated SNA dataset is enriched with diverse and up-to-date attack traffic samples, ensuring a balanced class distribution and including modern network protocols. Additionally, our research focuses on server-based dataset and capture real-time data to reflect the latest network traffic trends and include a comprehensive set of features to enhance the dataset's representation. In the realm of machine learning, researchers have employed a variety of approaches; however, they often don't strike a balance between computational cost and accuracy. In our study, we prioritize both accuracy and computational efficiency by introducing ensemble models.

III. PROPOSED METHODOLOGY

This study conducts experiments for server-based attack detection using machine learning. We implemented our experiments on a 12th-generation Intel Core i7 machine with 64 GB RAM, a 1TB SSD, and the Windows operating system. We utilized several other tools, including virtual machines (VMs), Kali Linux, and Wireshark for dataset collection. Furthermore, for attack detection, we deployed machine learning techniques using various libraries, such as Pandas, Scikit-learn, Keras, and TensorFlow.

Figure 1 illustrates the attack detection methodology. Initially, we collect a server-based attack dataset by setting up a network, as described in Section III-A. This dataset comprises three types of attacks: HTTP, SYN, and UDP. To prepare the data for machine learning models, we apply LabelEncoder to handle categorical features, converting them into numeric form.

Subsequently, we split the data into training and testing sets using an 80:20 ratio, with 80% allocated for model training and the remaining 20% for testing. We then introduce a novel stacked ensemble model, incorporating both base and meta-learners. To assess the performance of the proposed models, we evaluate them using the test set, measuring accuracy, precision, recall, and F1 score.

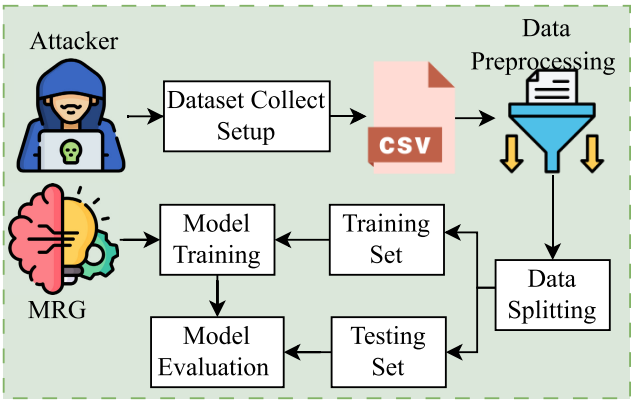

FIGURE 1. Proposed methodology implementation diagram.

A. NOVEL SNA DATASET COLLECTION METHODS

Our proposed methodology for creating a novel dataset for server-based network attacks is shown in Figure 2. We have established a virtual network using an Ubuntu Base Server, which hosts multiple services on virtual machines, such as FTP, SSH, HTTP, and DNS. To generate network attacks, we employ an attacker machine running Kali Linux. The attacker machine sends attack packets to the network hosted on the Ubuntu Base Server. The virtual network adapter within the network handles the reception and interaction of these attack packets. Each generated network attack traffic is captured using Wireshark, and the captured data is exported as pcapng files for further analysis. Pcapng files contain detailed information about network packets, including their source and destination addresses, timestamps, packet lengths, and protocol-specific data. Subsequently, the captured pcapng files are processed using ZUI software, converting them into Comma Separated Values (CSV) format files. Ultimately, we create a novel dataset based on network attack traffic, which can be utilized for subsequent predictive analysis.

The process of collecting our dataset involves the utilization of various techniques, tools, and methods. We outline each step involved in detail below:

- **Network Environment:** In this study, we established a virtual network on an Ubuntu-based server to capture attack traffic. Simulation technology enabled the hosting of multiple services on virtual machines (VMs), simulating real-world network scenarios. Wireshark, a widely-used network protocol analyzer, was employed for comprehensive monitoring and analysis of network

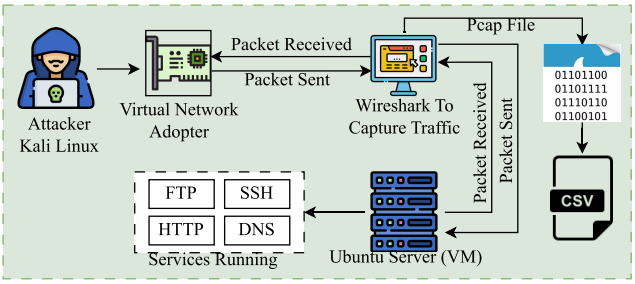

FIGURE 2. The novel dataset generation methodology.

traffic [31]. This configuration allowed for capturing and inspecting network packets across various protocols, providing valuable insights into potential vulnerabilities and security breaches.

This study generates network attacks using an attacker machine running Kali Linux. The attacker machine sends attack packets to a network of an Ubuntu Base Server. The study aimed to simulate real-world network attack scenarios by employing Kali Linux. The Ubuntu Base Server, serving as the target within the network, played a critical role in evaluating the impact of the generated network attacks. Using Kali Linux as the attacker machine and the Ubuntu Base Server as the target system ensured a controlled and reliable environment for conducting comprehensive network attack simulations.

- **Attacker Network:** Our research approach utilized Kali Linux, a powerful penetration testing platform known for its extensive array of security tools, to create an attack network. By employing the MHDDoS tool attack script [32], we simulated various DDoS attacks using an attacker machine Kali Linux, thereby generating our real-time network attack dataset. The MHDDoS tool attack script enables the generation of high-volume traffic to overwhelm target systems, effectively mimicking real-world attack scenarios. Additionally, incorporating the hping tool further enriched our dataset generation process [33], allowing for the customization of network packet characteristics and facilitating the emulation of sophisticated attack patterns. Below, we provide the commands utilized on the attacker's machine to generate attack network traffic:
  - hping -i u5000 -p 22 -S –rand-source 192.168.122.169
  - hping -S -p 4444 192.168.122.169
  - hping -S –flood -p 192.168.122.169
  - hping -2 u5000 -p 22 -S –rand-source 192.168.122.169
- **Virtual Network Adapter:** The virtual network adapter is responsible for receiving the attack packets and facilitating their interaction within the network [34]. Acting as a bridge between the physical network interface and the virtual network, the virtual network adapter receives incoming attack packets. It forwards them to the appropriate destinations within the network infrastructure. This vital component acts as a gateway, allowing the attack packets to traverse the network and interact with the various network devices, such as routers, switches, and servers.
- **Wireshark:** Our proposed research used Wireshark, a powerful open-source network protocol analyzer that enables network administrators and security professionals to capture and analyze network traffic [35]. Designed explicitly for Ubuntu-based servers, Wireshark provides comprehensive visibility into network activities, allowing users to monitor and troubleshoot network issues effectively by capturing and dissecting packets at various network stack layers. Each generated network attack traffic is captured by Wireshark and exported as pcapng file for further analysis, as illustrated in Figure 3.

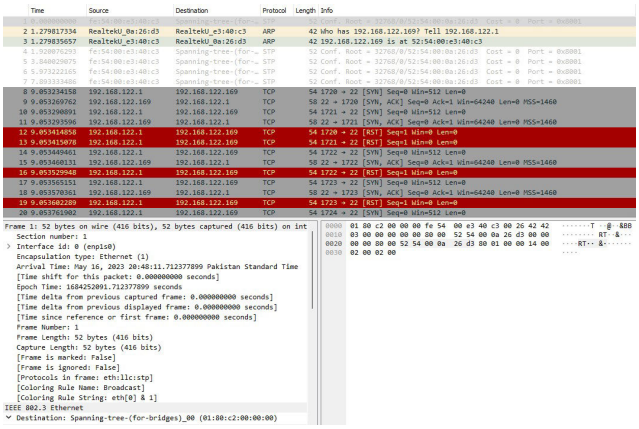

FIGURE 3. The Wireshark-based network traffic capture view.

- **Ubuntu Base Server:** We have used a Ubuntu Base Server, a popular choice for hosting various services on virtual machines, including file transfer (FTP), remote administration (SSH), web hosting (HTTP), and domain name resolution (DNS) [36]. These services serve essential functions in network environments. During network attacks by the attacker, services running on the Ubuntu Base Server, such as FTP, SSH, HTTP, and DNS, are susceptible to various security vulnerabilities. However, their widespread usage also makes them prime targets for malicious actors seeking to exploit vulnerabilities, gain unauthorized access, or disrupt system operations. The Ubuntu Base Server has a regularly monitored system log that can help detect and mitigate potential attacks targeting FTP, SSH, HTTP, and DNS services.
- **Dataset Formation:** our proposed research uses the Wireshark tool for network capturing, a popular network protocol analyzer tool that captures and stores network traffic in pcapng (Packet Capture Next Generation) files. Pcapng files contain detailed information about network packets, including their source and destination

addresses, timestamps, packet lengths, and protocol-specific data. The network traffic during attacks captured by the Wireshark tool is stored in pcapang files. The captured pcapng files are then input to the ZUI software for converting into Comma Separated Values(CSV) format files. During conversion, we also applied a data filter on extracted pcapng files, which includes the removal of "capture loss", "stats", and "reporter" variables. These filtered variables contain general information that may not be helpful for network attack detection. The formatted CSV files are captured related to each attack in our proposed research methodology. Table 2 describes the feature names in the dataset.

TABLE 2. The generated SNA dataset features description.

Feature	Feature	Feature	Feature
_path	ts	duration	uid
id.orig_p	id.resp_h	id.resp_p	proto
service	orig_bytes	resp_bytes	conn_state
missed_bytes	history	orig_pkts	orig_ip_bytes
resp_pkts	resp_ip_bytes	geo.orig.country_code	geo.orig.region
geo.orig.city	geo.orig.latitude	geo.orig.longitude	community_id
name	notice	peer	attack

Figure 4 illustrates the distribution of collected dataset samples for each attack. There are a total of 225,398 collected network traffic samples. The histogram chart shows that the dataset includes 125,959 UDP attack samples, 63,814 SYN attack samples, and 35,625 HTTP attack samples.

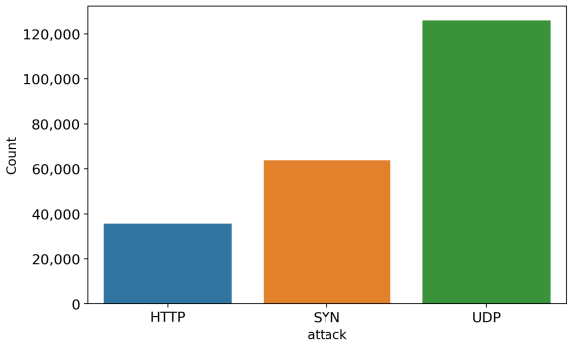

FIGURE 4. The target class distributions analysis.

B. AI-BASED ATTACK DETECTION MODELS

In our research, we utilized a diverse range of machine learning and deep learning models, each meticulously fine-tuned with its optimal hyperparameter configurations. The architecture details of the deep learning model are shown in Table 3. In our study, we explored three sequential models, namely CNN [37], GRU [38], and LSTM [39], for a multi-class classification task using Keras. The CNN model involves a Conv1D layer with 16 filters of size 3 and ReLU activation, followed by a MaxPooling1D layer with pool size 4. A flattened layer converts the output into a one-dimensional vector [40], and a Dropout layer with a rate of 0.6 mitigates overfitting [41]. The model concludes with a Dense layer with softmax activation for classification. The GRU model comprises 16 GRU units, a Dropout layer with a rate of 0.6, and a Dense output layer with softmax activation. Similarly, the LSTM model consists of 16 LSTM units, a Dropout layer with a rate of 0.6, and a Dense output layer with softmax activation. All models are compiled with categorical cross-entropy loss and optimized using the Adam optimizer [42], [43]. During training, a validation split of 0.1 is employed, and the models are trained for 15 epochs. The objective is to minimize the categorical cross-entropy loss and enable accurate multi-class classification for the network attack detection task.

TABLE 3. Deep learning models architectures.

CNN	GRU	LSTM
Sequential() Conv1D(16, 3, input_shap=-, activation='relu') MaxPool1D (pool_size=(4)) Flatten() Dropout(0.6)	Sequential() (16, input_shape) Dropout(0.6)	Sequential() LSTM(16, input_shape) Dropout(0.6)
Dense(3,activation='softmax') compile(loss = 'categorical_crossentropy',optimizer = 'adam') fit(validation_split=0.1, epochs=15)		

Additionally, we employ four machine learning models—LR, SVM, RF, and GNB—utilizing the optimal hyperparameter settings outlined in Table 4. These parameter configurations play a pivotal role in fine-tuning the models for substantial performance gains while preventing overfitting on the training data. The selection of these hyperparameter settings involves a meticulous grid-search method to ensure an effective balance. Through careful selection and optimization of these hyperparameters, our goal is to improve the accuracy and generalization capabilities of our models. This enhancement enables them to effectively detect network attacks and provide robust security measures.

TABLE 4. Machine learning models hyperparameter settings.

Models	Hyperparameter Values
LR	random_state= 0, solver= 'lbfgs', max_iter= 90, multi_class= 'auto', C= 1.0
SVM	random_state= 0, max_iter= 500, penalty= 'l2', loss= 'squared_hinge', tol= 1e-4
GNB	var_smoothing= 1e-9, priors= None
RF	criterion= 'entropy', max_depth= 2, min_samples_split= 2, min_samples_leaf= 1, max_features= "sqrt", bootstrap= True, n_estimators= 1, random_state= 0

1) META-RF-GNB (MRG) MODEL FOR NETWORK ATTACK DETECTION

In comparison with state-of-the-art models, we propose a novel model named the MRG for network attack detection, harnessing the potential of ensemble learning through model stacking [44]. MRG combines two distinct base models:

GNB and RF, to enhance the accuracy and robustness of network security systems. Additionally, it introduces a meta-learner, another RF, that significantly amalgamates the outputs of these base-level models to make a final prediction as illustrated in Figure 5. We selected these two models for our MRG approach because of their significant performance at the individual level. The synergy of these components not only improves detection accuracy but also enhances the robustness of network security systems in the face of evolving threats. Further, we provide an in-depth understanding of MRG, including the mathematical foundations of its constituent models and the mechanisms through which they collaborate.

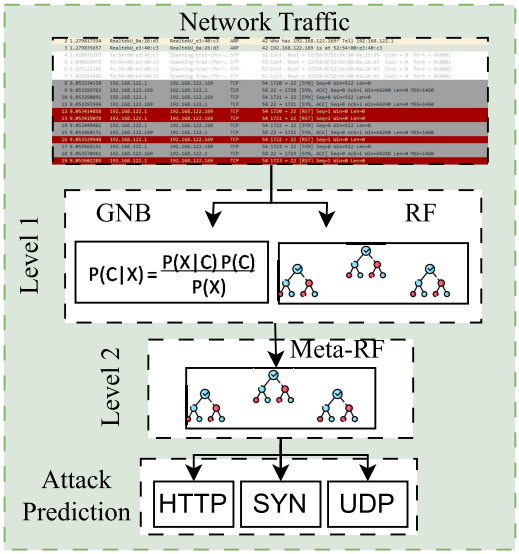

FIGURE 5. The architecture of the novel proposed MRG approach.

We use GNB as a base learner which is a probabilistic classification algorithm that makes predictions based on Bayes' theorem. It assumes that the features are conditionally independent, given the class label. Mathematically, GNB computes the class probability $P(C_k|\mathbf{x})$ for a data instance $\mathbf{x}$ belonging to class C_k as:

$$P(C_k|\mathbf{x}) = \frac{P(C_k) \cdot P(\mathbf{x}|C_k)}{P(\mathbf{x})}$$

where:

- $P(C_k)$ is the prior probability of class C_k.
- $P(\mathbf{x}|C_k)$ is the likelihood of the data given the class.
- $P(\mathbf{x})$ is the probability of observing data $\mathbf{x}$.

While RF is also used as a base learner and meta-learner, it is an ensemble learning technique that combines multiple decision trees to make predictions. The prediction of an RF model for a given instance $\mathbf{x}$ is an aggregation of the predictions from its decision trees. Mathematically, the RF prediction can be represented as:

$$\hat{y}_{\text{RF}} = \frac{1}{N} \sum_{i=1}^{N} h_i(\mathbf{x})$$

where

- $\hat{y}_{\text{RF}}$ is the RF prediction.
- N is the number of decision trees in the RF.
- $h_i(\mathbf{x})$ is the prediction of the i-th decision tree for instance $\mathbf{x}$.

The key contribution of MRG lies in its model stacking approach. For each network traffic instance, both GNB and RF generate predictions ($\hat{y}_{\text{GNB}}$ and $\hat{y}_{\text{RF}}$). These predictions are then stacked into a vector $\mathbf{X}_{\text{stacked}}$. The meta-learner, typically another RF, takes $\mathbf{X}_{\text{stacked}}$ as input and learns how to make the final prediction $\hat{y}_{\text{MRG}}$. This process can be mathematically represented as:

$$\mathbf{X}_{\text{stacked}} = [\hat{y}_{\text{GNB}}, \hat{y}_{\text{RF}}]$$

By stacking models at the base level, MRG harnesses the unique strengths of both GNB and RF. GNB, known for its simplicity and ability to handle continuous and discrete data, complements RF's prowess in capturing complex relationships and feature importance. This duality enables MRG to adapt to a wide range of network traffic scenarios effectively. Algorithm 1, shows the steps for MGR. For each instance in the test dataset, MRG first obtains GNB and RF predictions, represented as $y\hat{\text{GNB}}$ and $y\hat{\text{RF}}$, respectively. These predictions are stacked together and provided as input to the meta-learner, which is another RF model. The meta-learner, denoted as Meta-Learner, combines the base models' outputs to produce a final prediction, $y\hat{\text{MRG}}$, indicating whether the network traffic instance contains an attack.

Algorithm 1 Meta-RF-GNB (MRG) for Network Attack Detection

Input: Training data D, Test data D_{test}
Train Gaussian Naive Bayes (GNB) model on D
Train Random Forest (RF) model on D
Initialize Meta-Learner model
for each instance $\mathbf{x}$ in D_{test} **do**
  $\hat{y}_{\text{GNB}} = \text{GNB}(\mathbf{x})$ {Make a GNB prediction}
  $\hat{y}_{\text{RF}} = \text{RF}(\mathbf{x})$ {Make an RF prediction}
  $\mathbf{X}_{\text{stacked}} = [\hat{y}_{\text{GNB}}, \hat{y}_{\text{RF}}]$ {Combine GNB and RF predictions}
  $\hat{y}_{\text{MRG}} = \text{Meta-Learner}(\mathbf{X}_{\text{stacked}})$ {Use the meta-learner to make the final prediction}
  Store $\hat{y}_{\text{MRG}}$ as the final prediction for $\mathbf{x}$ {Final prediction for the current instance}
end for
Output: Final predictions for all instances in D_{test}

IV. RESULTS

In this section, we provide an analysis of the machine learning and deep learning models' performance on our collected SNA dataset. The evaluation metrics used to measure the effectiveness of these models include precision, recall, and F1 score [18]. These performance parameters can be

defined as follows:

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

Here, TP stands for True Positives (correctly identified positive instances), TN stands for True Negatives (correctly identified negative instances), FP stands for False Positives (incorrectly identified as positive instances), and FN stands for False Negatives (incorrectly identified as negative instances).

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

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

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

A. RESULTS USING MACHINE LEARNING MODELS

Table 5 displays the comparative performance outcomes of various applied machine learning techniques. Our study incorporates advanced state-of-the-art models, including RF, LR, SVM, and GNB. The analysis reveals that LR and SVM achieved relatively lower performance scores in comparison. In contrast, RF and GNB demonstrated commendable performance scores, both achieving an accuracy of 0.9890, although not the highest among the evaluated models. RF displayed exceptional precision, recall, and F1 scores for all target classes, resulting in an overall average of 0.99. On the other hand, LR and SVM exhibited lower performance scores with an accuracy of 0.9032 for LR and 0.8966 for SVM. LR showed varied precision, recall, and F1 scores for different target classes, resulting in an average of 0.90. Similarly, SVM demonstrated varying performance for different target classes. Both RF and GNB exhibited better robustness to noisy data, effectively handling data imperfections and variations, which is advantageous in real-world scenarios where data may not be perfectly clean. RF's feature selection capability improved model performance by focusing on the most informative features, while GNB's probabilistic approach, assuming that features are conditionally independent, simplified the modeling process and contributed to accurate predictions.

The confusion matrix analysis of machine learning models is visualized in Figure 6. The findings reveal that the SVM model exhibits a relatively high ratio of incorrect predictions in network attack classifications, while the GNB model provides a high correct prediction ratio. GNB gives us 44,592 correct predictions and 488 wrong predictions out of a total of 45,080.

The results of our meta-learning MRG approach, shown in Table 6, highlight its outstanding performance. We achieved a remarkable accuracy of 0.9999 in network attack detection, with perfect precision, recall, and F1 scores for each attack category, such as HTTP, SYN, and UDP. The significant performance is attributed to the two-level architecture in the stack, leveraging the synergy of two individual models. GNB excels with its probabilistic power, while RF demonstrates its tree-based prediction capabilities. Combining both models allows us to harness the full power of their strengths, resulting in significant improvements across all evaluation metrics.

TABLE 5. Performance analysis of machine learning models for testing data.

Model	Accuracy	Attack	Precision	Recall	F1
RF	0.9890	HTTP	1.00	1.00	1.00
		SYN	1.00	0.96	0.98
		UDP	0.98	1.00	0.99
		Average	0.99	0.99	0.99
LR	0.9032	HTTP	0.76	0.56	0.65
		SYN	1.00	1.00	1.00
		UDP	0.88	0.95	0.92
		Average	0.90	0.90	0.90
SVM	0.8966	HTTP	1.00	0.35	0.52
		SYN	1.00	1.00	1.00
		UDP	0.84	1.00	0.92
		Average	0.91	0.90	0.88
GNB	0.9890	HTTP	1.00	1.00	1.00
		SYN	1.00	0.96	0.98
		UDP	0.98	1.00	0.99
		Average	0.99	0.99	0.99

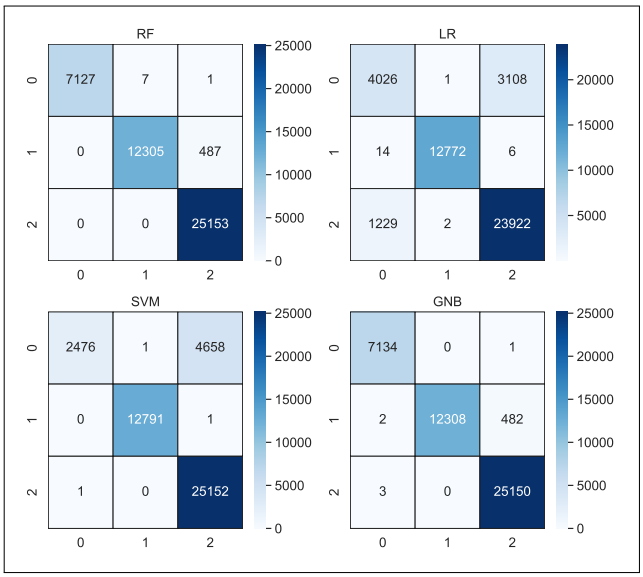

FIGURE 6. Confusion matrix for machine learning models.

TABLE 6. Performance analysis of novel meta-learning-based MRG model for testing data.

Model	Accuracy	Attack	Precision	Recall	F1
MRG	0.9999	HTTP	1.00	1.00	1.00
		SYN	1.00	1.00	1.00
		UDP	1.00	1.00	1.00
		Average	1.00	1.00	1.00

Our MRG approach's confusion matrix, as shown in Figure 7, illustrates the significant performance and reliability of our model in real-world network security applications. Notably, our proposed models exhibit an exceptionally low wrong prediction ratio, with only one incorrect prediction out

of 45,080 total predictions. There are no false negatives for any class (1, 2, and 3), resulting in a false negative rate of 0 for our proposed MRG. However, the false positive rate for Class 1 is 1, indicating that the model falsely identified instances as Class 1. Despite this, the overall low rates of false positives and negatives underscore the significance of our proposed models.

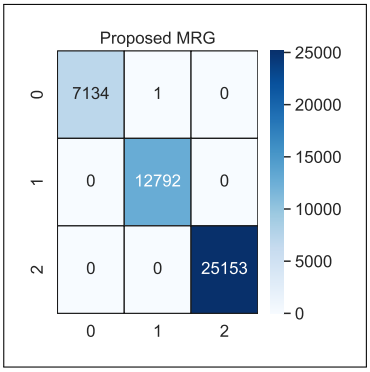

FIGURE 7. Confusion matrix for novel meta-learning-based MRG model.

The machine learning methods have been assessed using a 10-fold cross-validation approach, as shown in Table 7. In this k-fold validation process, we have divided the novel dataset into ten equal parts, allowing for comprehensive validation. The results indicate that RF and GNB have achieved a strong mean accuracy of 0.9812 and 0.9867, respectively, confirming the effectiveness and generalizability of these methods. In contrast, LR and SVM have demonstrated lower performance with accuracy values of 0.9001 and 0.9238, respectively, and have exhibited relatively higher standard deviations (SD), indicating variations in performance across the folds. Comparing our approach, MRG has outperformed other models with a mean accuracy score of 0.9994 and a minimal SD of 0.0002. These results underscore the significance of our proposed approach and demonstrate its resistance to overfitting.

TABLE 7. Machine learning results using k-fold cross validation.

Model	Accuracy	SD (+/-)
RF	0.9812	0.0008
LR	0.9001	0.0220
SVM	0.9238	0.0250
GNB	0.9867	0.0035
MRG	0.9994	0.0002

B. RESULTS USING DEEP LEARNING MODELS

In comparison to machine learning models, we also evaluated deep learning models. Table 8 presents the performance analysis of deep learning models on the testing data. Each model, including CNN, LSTM, and GRU, achieved exceptionally high accuracy, precision, recall, and F1 scores across different attack categories (HTTP, SYN, and UDP). The "Average" row underscores the consistent and strong performance of all models, with perfect scores of 1.00, highlighting their reliability in network attack classification. Notably, the GRU model stands out with the highest accuracy of 0.9998, affirming its effectiveness in real-world network security applications. This outstanding performance of the GRU model can be attributed to its recurrent architecture, which is more adept at capturing sequential patterns. GRUs excel in capturing long-range dependencies in the data, making them particularly effective for tasks involving time series or sequential data, such as network traffic behavior. The performance of the deep learning models is significant when compared to individual machine learning models, but our proposed MRG model also demonstrates its significance, especially in terms of accuracy scores.

TABLE 8. Performance analysis of deep learning models for testing data.

Model	Accuracy	Attack	Precision	Recall	F1
CNN	0.9987	HTTP	1.00	1.00	1.00
		SYN	1.00	1.00	1.00
		UDP	1.00	1.00	1.00
		Average	1.00	1.00	1.00
LSTM	0.9997	HTTP	1.00	1.00	1.00
		SYN	1.00	1.00	1.00
		UDP	1.00	1.00	1.00
		Average	1.00	1.00	1.00
GRU	**0.9998**	HTTP	1.00	1.00	1.00
		SYN	1.00	1.00	1.00
		UDP	1.00	1.00	1.00
		Average	**1.00**	**1.00**	**1.00**

Table 9 presents the performance analysis of deep learning models over different epochs during training. For the CNN, we observed a gradual improvement in performance from epoch 1 to epoch 5. The training accuracy increased from 0.7624 to 0.9586, while the validation accuracy reached 0.9988. Similarly, the GRU and LSTM models also demonstrated significant enhancements in performance with each epoch. For GRU, the training accuracy improved from 0.8568 to 0.9607, and the validation accuracy reached 0.9999. The LSTM model exhibited similar trends, with the training accuracy increasing from 0.9622 to 0.9958, and the validation accuracy reaching 0.9998. The performance improvements across epochs showcase the ability of these deep learning models to learn and adapt to the dataset, resulting in high accuracy for network attack detection.

Table 10 presents the performance validation analysis of deep learning methods using the k-fold approach. The analysis was conducted with ten folds of data for k-fold cross-validation. The results indicate that the CNN method achieved an average performance score of 0.9660 with a relatively high SD of 0.0649. In contrast, the LSTM and GRU methods demonstrated excellent k-fold scores, indicating their ability to generalize well for network attack detection across different data portions. These findings highlight the effectiveness of LSTM and GRU in handling variations and complexities in the dataset, making them promising choices for network intrusion detection.

TABLE 9. Per epochs performance analysis of deep learning models.

Epoch	Train Loss	Train Acc.	Val. Loss	Val. Acc.
CNN				
1	264.09	0.7624	1.8846	0.9301
2	0.7565	0.8929	0.1658	0.9118
3	0.2240	0.9172	0.1187	0.9442
4	0.1691	0.9431	0.0940	0.9453
5	0.1278	0.9586	0.0436	0.9988
GRU				
1	0.3783	0.8568	0.1550	0.9458
2	0.2089	0.9224	0.0177	0.9992
3	0.1133	0.9563	0.0022	0.9998
4	0.1032	0.9568	0.0020	0.9999
5	0.0997	0.9607	0.0009	0.9999
LSTM				
1	0.1026	0.9622	0.0022	0.9998
2	0.0264	0.9929	0.0028	0.9998
3	0.0137	0.9958	0.0030	0.9998
4	0.0909	0.9821	0.0038	0.9997
5	0.0287	0.9931	0.0041	0.9996

TABLE 10. Performance validation analysis of applied deep learning approaches.

Model	Accuracy	SD (+/-)
CNN	0.9660	0.0649
LSTM	0.9990	0.0001
GRU	0.9991	0.0010

The confusion matrices of deep learning techniques are shown in Figure 8. The results highlight the deep learning approach exhibits a lower rate of wrong predictions, indicative of its superior performance. Notably, the proposed GRU model stands out with remarkable accuracy, making only 8 incorrect predictions out of 45,080 test predictions, while correctly classifying 45,072 instances in unseen testing data.

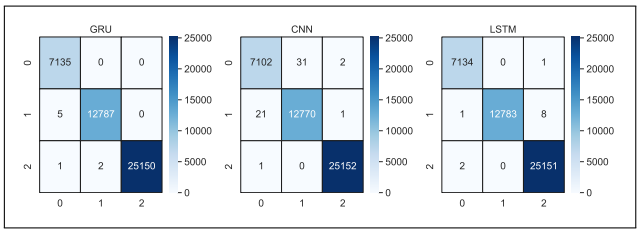

FIGURE 8. Confusion matrices for deep learning models.

The violin plot visualizes the distribution of accuracy scores for various machine learning and deep learning models, such as RF, LR, SVM, GNB, MRG, CNN, LSTM, and GRU. The plot reveals subtle differences in their performance as shown in Figure 9. RF, GNB, and MRG exhibit comparable accuracy scores, with RF at 0.9890 and GNB at 0.9890, indicating good performance. LR and SVM show slightly lower accuracy, scoring 0.9032 and 0.8966, respectively. MRG stands out with the highest accuracy of 0.9999, significantly narrower distribution, and superior performance. CNN and LSTM also performed well, scoring 0.9987 and 0.9997, respectively. GRU performs closely to MRG with an accuracy of 0.9998 but has a slightly broader distribution, suggesting a minor difference in performance.

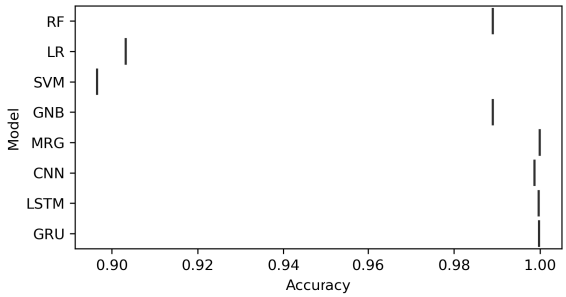

FIGURE 9. Accuracy comparison using violin plot.

For computational complexity analysis of both deep learning and machine learning methods, we measured computation time in seconds (S), which is presented in Figure 10. The results reveal insights into the computational demands of these models for network attack detection.

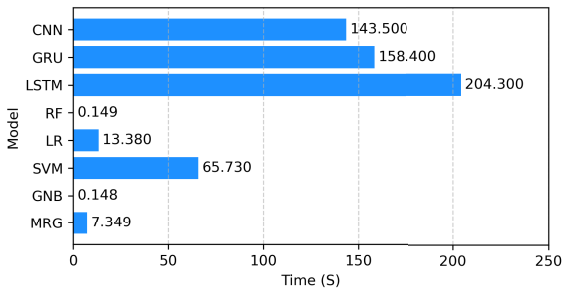

FIGURE 10. Computational complexity analysis.

The deep learning models, including CNN, GRU, and LSTM, exhibit varying levels of computational complexity. Among them, LSTM stands out with the highest computational demand, while both CNN and GRU demonstrate lower computational complexity in comparison, though they are still considered relatively high. On the other hand, machine learning models, such as RF, LR, SVM, and GNB, also show differences in computational complexity. SVM exhibits the highest computational complexity, followed by LR. In contrast, RF and GNB are notably more efficient in terms of runtime execution, requiring significantly less time for computation. These computational insights provide valuable considerations when selecting models for network attack detection based on the available computational resources.

C. EXPLAINABLE ARTIFICIAL INTELLIGENCE (XAI) ANALYSIS

The Shapley Additive Explanations (SHAP) chart-based Explainable AI (XAI) analysis of the Meta-learner Random Forest method is illustrated in Figure 11. For the XAI analysis, we have selected the top 7 network dataset features.

This research involves dissecting the meta-learner Random Forest to reveal the features and patterns influencing its decisions, shedding light on the factors contributing to detecting or misclassifying network attacks. This analysis concludes that the network features id.resp_p, resp_bytes, proto, ts, id.orig_p, id.orig_h, and resp_ip_bytes have a high involvement during network attack detection.

D. PERFORMANCE COMPARISON WITH OTHER METHODS

In this section, we conducted a comparative analysis of our approach alongside other methodologies employed in related studies. To ensure fair performance evaluation, we implemented these methods using our novel dataset within the same experimental environment. We selected recent studies in the field of intrusion detection and applied their architectures to our dataset. The results are presented in Table 11. The study by [45] utilizes BayseNet for intrusion detection in SDN networks. In a similar vein, [46] deploys Artificial Neural Networks (ANN) for DDoS attack detection. Likewise, [47] focuses on DDoS attack detection in Internet service provider networks. The study presented by [48] proposes a cluster-based semi-supervised approach for DDoS attack prediction using RF. The analysis demonstrates that our proposed approach outperformed state-of-the-art methods in network attack detection, achieving higher performance scores. The significant advantage of our proposed approach lies in its simplicity for decision-making in attacks. In contrast to previous studies, where researchers often employed complex neural networks, such as in [46], or utilized optimizers that could escalate computational costs [26], our approach stands out. We leverage state-of-the-art, low computational cost algorithms arranged in a hierarchical manner, achieving both significant accuracy and efficiency.

TABLE 11. Performance comparisons of our proposed approach with state-of-the-art studies.

Ref.	Year	Method	Accuracy
[45]	2016	BayesNet	0.916
[46]	2019	ANN	0.843
[47]	2020	XGBoost	0.980
[48]	2021	RF	0.966
[18]	2022	RF+(SVD,PCA)	0.981
[26]	2023	Ensemble+PSO	0.988
Our	2023	**Novel MRG**	**0.9999**

E. VALIDATION OF PROPOSED MRG USING BENCHMARK DATASETS

To validate the generalization of the novel proposed MRG approach, we have implemented it on benchmark datasets such as CICIDS2017 [49] and CSECICIDS2018 [50]. The CICIDS2017 dataset characterized the abstract behavior of 25 users by examining their activities across protocols such as HTTP, SSH, FTP, HTTPS, and email. Similarly, it was constructed by analyzing protocols like HTTPS, POP3, IMAP, HTTP, SSH, SMTP, and FTP. We deploy the MRG approach on these datasets and the results are shown in Table 12. The analysis shows that our proposed MRG approach achieved high-performance accuracy scores for both datasets, highlighting its significance in network attack detection.

TABLE 12. The performance validation of proposed MRG model using benchmark datasets.

Dataset	Accuracy	Precision	Recall	F1
CICIDS2017	0.98	0.98	0.98	0.98
CSECICIDS2018	0.99	0.99	0.99	0.99

F. DISCUSSION

In this research, we introduce the SNA dataset, a novel and valuable resource for network intrusion detection. Unlike many existing benchmark datasets, SNA is generated from real-world server-based networks, reflecting current traffic patterns and modern protocols. The dataset's significance lies in its real-world relevance, uniqueness, customization, impact assessment, and contribution to research.

SNA's real-world relevance is pivotal for understanding contemporary network threats. Its unique generation process introduces novel traffic patterns and scenarios, fostering innovative research in intrusion detection and network security. With Kali Linux as the attacker machine and specific attack types targeting an Ubuntu Base Server, SNA enables tailored attack scenarios, offering insights into security vulnerabilities. By targeting critical services like FTP, SSH, HTTP, and DNS, SNA facilitates the evaluation of real-world attack impact on vital network services. It augments the research landscape by serving as a resource to assess and benchmark intrusion detection systems and security algorithms, advancing the development of effective security solutions.

In regards to the machine learning approach, we work on accuracy and as well as efficient models in terms of computational cost. In our comparative analysis of various models, we found that MRG stands out as a promising approach. It not only achieved a remarkable accuracy of 99.999%, indicating its effectiveness in accurately detecting network attacks but also demonstrated efficiency in terms of computational cost as shown in Figure 12. This combination of high accuracy and low computational cost makes MRG a compelling choice for network intrusion detection tasks.

Furthermore, the deep learning model GRU and machine learning model MRG perform very closely, with the primary difference lying in their computational cost. To investigate the significance of the proposed MRG model over the GRU model, we conducted a statistical t-test [51]. We combined all the results of MRG, including computational cost, into one sample, and did the same for GRU. The t-test results indicate that there is no significant difference between the two models.

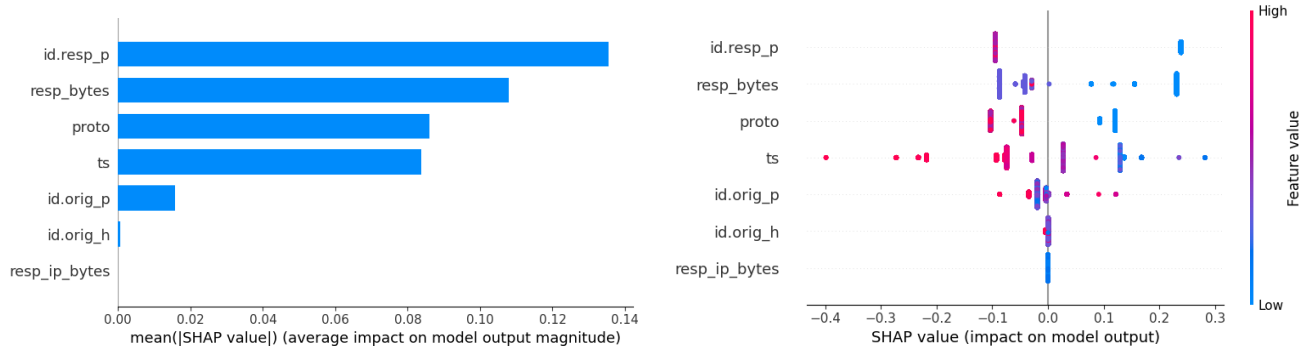

FIGURE 11. The SHAP chart-based XAI analysis of Meta learner Random Forest method using top 7 network features.

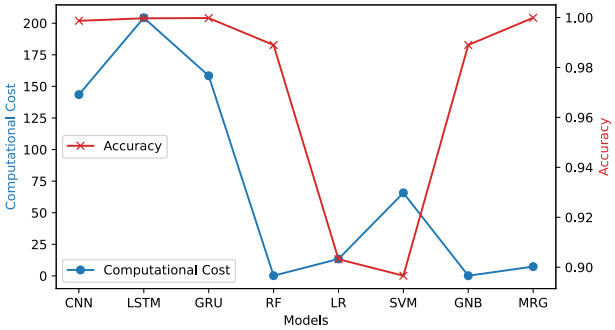

FIGURE 12. Computational Cost vs Accuracy.

However, at a significance level ($\alpha = 0.05$), MRG shows statistical significance over GRU, with a t-statistic score of 0.96 and a p-value of 0.34. This difference could be attributed to the substantial variation in computational cost.

V. CONCLUSION AND FUTURE WORK

This research introduced a novel methodology for generating the SNA dataset, a unique network intrusion dataset. By utilizing simulation technology on an Ubuntu Base Server and Kali Linux as the attacker machine, we replicated real-world network scenarios, capturing attack traffic. Our collected SNA dataset represents a significant improvement over existing benchmark datasets due to its specification in HTTP, UDP, and SYN attacks on the server. It incorporates real-time traffic with the latest protocols and leverages state-of-the-art machine learning models, resulting in remarkable accuracy scores. Notably, GRU outperformed other machine learning models with an accuracy score of 0.9998, while LSTM and CNN also achieved high accuracy at 0.9997. Importantly, our proposed models achieved results comparable to deep learning models but with lower computational costs. Our MRG model achieved a remarkable accuracy score of 0.9999 with only 7.349 seconds of computation time. All these findings emphasize the significance of stacked and meta-learning architecture over others in terms of both computational cost and accuracy.

A. LIMITATIONS AND FUTURE WORK

Despite its significance, our study has limitations. SNA's data collection is confined to local area networks, potentially restricting its representation of diverse and larger-scale network environments. The dataset's specificity to Ubuntu-based systems may limit its generalizability to other server platforms or operating systems. Additionally, SNA's size may be constrained for handling extensive network traffic scenarios. Future work will focus on enhancing dataset scalability for larger-scale scenarios and exploring its adaptability to different server platforms and operating systems. Ongoing research will delve into novel attack vectors and scenarios to advance our understanding of network security.

DATA AVAILABILITY STATEMENT

The dataset resulting from this study, titled 'Server-based Network Attack (SNA) dataset,' is available on Mendeley Data and accessible through the DOI: 10.17632/g3pcpm92jr.1.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] S. Ashfaq, Y. Tang, and R. Maqbool, "Insights of energy and its trade networking impacts on sustainable economic development," *Energy*, vol. 265, Feb. 2023, Art. no. 126319.

[2] F. Rustam, A. Raza, I. Ashraf, and A. D. Jurcut, "Deep ensemble-based efficient framework for network attack detection," in *Proc. 21st Medit. Commun. Comput. Netw. Conf. (MedComNet)*, Jun. 2023, pp. 1–10.

[3] A. Gupta, B. D. Mazumdar, M. Mishra, P. P. Shinde, S. Srivastava, and A. Deepak, "Role of cloud computing in management and education," *Mater. Today, Proc.*, vol. 80, pp. 3726–3729, Sep. 2023.

[4] L. Cao, X. Jiang, Y. Zhao, S. Wang, D. You, and X. Xu, "A survey of network attacks on cyber-physical systems," *IEEE Access*, vol. 8, pp. 44219–44227, 2020.

[5] P. Kumari and A. K. Jain, "A comprehensive study of DDoS attacks over IoT network and their countermeasures," *Comput. Secur.*, vol. 127, Apr. 2023, Art. no. 103096.

[6] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments," *Energy Rep.*, vol. 7, pp. 8176–8186, Nov. 2021.

[7] L. Irwin. *List of Data Breaches and Cyber Attacks in 2023*. Accessed: Jul. 19, 2023. [Online]. Available: https://www.itgovernance.co.uk/blog/list-of-data-breaches-and-cyber-attacks-in-2023
[8] A. T. Tunggal. *What is the Cost of a Data Breach in 2023?* Accessed: Jul. 19, 2023. [Online]. Available: https://www.upguard.com/blog/cost-of-data-breach
[9] M. Sikora, R. Fujdiak, and J. Misurec, "Analysis and detection of application-independent slow denial of service cyber attacks," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Nov. 2021, pp. 1–6.
[10] A. Raza, K. Munir, M. S. Almutairi, and R. Sehar, "Novel class probability features for optimizing network attack detection with machine learning," *IEEE Access*, vol. 11, pp. 98685–98694, 2023.
[11] M. Imran, H. U. R. Siddiqui, A. Raza, M. A. Raza, F. Rustam, and I. Ashraf, "A performance overview of machine learning-based defense strategies for advanced persistent threats in industrial control systems," *Comput. Secur.*, vol. 134, Nov. 2023, Art. no. 103445.
[12] A. Arshad, M. Jabeen, S. Ubaid, A. Raza, L. Abualigah, K. Aldiabat, and H. Jia, "A novel ensemble method for enhancing Internet of Things device security against botnet attacks," *Decis. Analytics J.*, vol. 8, Sep. 2023, Art. no. 100307.
[13] M. M. Raikar and S. M. Meena, "SSH brute force attack mitigation in Internet of Things (IoT) network : An edge device security measure," in *Proc. 2nd Int. Conf. Secure Cyber Comput. Commun. (ICSCCC)*, May 2021, pp. 72–77.
[14] D. S. M. Gonçalves, R. S. Couto, and M. G. Rubinstein, "A protection system against HTTP flood attacks using software defined networking," *J. Netw. Syst. Manage.*, vol. 31, no. 1, p. 16, Jan. 2023.
[15] B. Rajendran, G. Palaniappan, R. Dijesh, B. S. Bindhumadhava, and S. D. Sudarsan, "A universal domain name resolution service—Need and challenges—Study on blockchain based naming services," in *Proc. IEEE Region 10th Symp. (TENSYMP)*, Jul. 2022, pp. 1–6.
[16] S. A. M. Al-Juboori, F. Hazzaa, Z. S. Jabbar, S. Salih, and H. M. Gheni, "Man-in-the-middle and denial of service attacks detection using machine learning algorithms," *Bull. Electr. Eng. Informat.*, vol. 12, no. 1, pp. 418–426, Feb. 2023.
[17] J. Hancock, T. M. Khoshgoftaar, and J. L. Leevy, "Detecting SSH and FTP brute force attacks in big data," in *Proc. 20th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2021, pp. 760–765.
[18] F. Rustam, M. Mushtaq, A. Hamza, M. Farooq, A. Jurcut, and I. Ashraf, "Denial of service attack classification using machine learning with multi-features," *Electronics*, vol. 11, no. 22, p. 3817, Nov. 2022.
[19] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives," in *Proc. IEEE 3rd Int. Conf. Comput., Commun. Secur. (ICCCS)*, Oct. 2018, pp. 1–8.
[20] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.
[21] J. Song, H. Takakura, and Y. Okabe. (2006). *Description of Kyoto University Benchmark Data*. Accessed: Mar. 15, 2016. [Online]. Available: http://www.takakura.com/Kyoto_data/BenchmarkData-Description-v5.pdf
[22] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, May 2012.
[23] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.
[24] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, vol. 1, 2018, pp. 108–116.
[25] (2018). *IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*. Accessed: Jun. 24, 2023. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html
[26] F. Rustam and A. D. Jurcut, "Malicious traffic detection in multi-environment networks using novel S-DATE and PSO-D-SEM approaches," *Comput. Secur.*, vol. 136, Jan. 2024, Art. no. 103564.
[27] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly based network intrusion detection for IoT attacks using deep learning technique," *Comput. Electr. Eng.*, vol. 107, Apr. 2023, Art. no. 108626.
[28] D. Kilichev and W. Kim, "Hyperparameter optimization for 1D-CNN-based network intrusion detection using GA and PSO," *Mathematics*, vol. 11, no. 17, p. 3724, Aug. 2023.
[29] F. Rustam, A. D. Jurcut, W. Aljedaani, and I. Ashraf, "Securing multi-environment networks using versatile synthetic data augmentation technique and machine learning algorithms," in *Proc. 20th Annu. Int. Conf. Privacy, Secur. Trust (PST)*, Aug. 2023, pp. 1–10.
[30] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: A deep learning-based intrusion detection framework for securing IoT," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, Mar. 2022, Art. no. e3803.
[31] A. Nath, *Packet Analysis With Wireshark*. Birmingham, U.K.: Packt Publishing, 2015.
[32] W. Yunanri, Y. Yuliadi, S. Esabella, and Y. B. Fitriana, "Analisis keamanan jaringan menggunakan metode security policy development life cycle (SPDLC)," *KLIK, Kajian Ilmiah Informatika Komputer*, vol. 4, no. 1, pp. 634–641, 2023.
[33] K. Org. *Hping3 Documentation*. Accessed: Nov. 10, 2023. [Online]. Available: https://www.kali.org/tools/hping3/
[34] Z. Xiang and P. Seeling, "Mininet: An instant virtual network on your computer," in *Computing in Communication Networks*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 219–230.
[35] K. Cardwell, "Advanced features of Wireshark," in *Tactical Wireshark: A Deep Dive Into Intrusion Analysis, Malware Incidents, and Extraction of Forensic Evidence*. Berkeley, CA, USA: Springer, 2023, pp. 183–219.
[36] A. Kouka, *Ubuntu Server Essentials*. Birmingham, U.K.: Packt Publishing, 2015.
[37] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1222–1228.
[38] B. Cao, C. Li, Y. Song, Y. Qin, and C. Chen, "Network intrusion detection model based on CNN and GRU," *Appl. Sci.*, vol. 12, no. 9, p. 4184, Apr. 2022.
[39] S. A. Althubiti, E. M. Jones, and K. Roy, "LSTM for anomaly-based network intrusion detection," in *Proc. 28th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2018, pp. 1–3.
[40] A. A. Reshi, F. Rustam, A. Mehmood, A. Alhossan, Z. Alrabiah, A. Ahmad, H. Alsuwailem, and G. S. Choi, "An efficient CNN model for COVID-19 disease detection based on X-ray image classification," *Complexity*, vol. 2021, pp. 1–12, May 2021.
[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Sep. 2014.
[42] A. Semenov, V. Boginski, and E. L. Pasiliao, "Neural networks with multidimensional cross-entropy loss functions," in *Proc. 8th Int. Conf. Comput. Data Social Netw.* Cham, Switzerland: Springer, 2019, pp. 57–62.
[43] I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, "Adam optimization algorithm for wide and deep neural network," *Knowl. Eng. Data Sci.*, vol. 2, no. 1, pp. 41–46, Jun. 2019.
[44] B. Pavlyshenko, "Using stacking approaches for machine learning models," in *Proc. IEEE 2nd Int. Conf. Data Stream Mining Process. (DSMP)*, Aug. 2018, pp. 255–258.
[45] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, and B. Yang, "Predicting network attack patterns in SDN using machine learning approach," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2016, pp. 167–172.
[46] A. Sanmorino, "A study for DDOS attack classification method," *J. Phys., Conf. Ser.*, vol. 1175, Mar. 2019, Art. no. 012025.
[47] N. N. Tuan, P. H. Hung, N. D. Nghia, N. V. Tho, T. V. Phan, and N. H. Thanh, "A DDoS attack mitigation scheme in ISP networks using machine learning based on SDN," *Electronics*, vol. 9, no. 3, p. 413, Feb. 2020.
[48] M. Aamir and S. M. A. Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 33, no. 4, pp. 436–446, May 2021.
[49] x. Chen, (2023), "CICIDS2017 and UNBSW-NB15," *IEEE DataPort*, doi: 10.21227/ykpn-jx78.
[50] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "A detailed analysis of the CICIDS2017 data set," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, Funchal-Madeira, Portugal. Cham, Switzerland: Springer, 2018, pp. 172–188.
[51] T. K. Kim, "T test as a parametric statistic," *Korean J. Anesthesiol.*, vol. 68, no. 6, pp. 540–546, 2015.

FURQAN RUSTAM received the M.C.S. degree from the Department of Computer Science, The Islamia University of Bahawalpur, Pakistan, in October 2017, and the Master of Computer Science degree from the Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology (KFUEIT), Rahim Yar Khan, Pakistan. He is currently pursuing the Ph.D. degree in computer science with University College Dublin, Ireland. He was a Research Assistant with the Fareed Computing & Research Center, KFUEIT. His recent research interests include data mining, machine learning, and artificial intelligence, mainly working on creative computing and supervised machine learning.

ALI RAZA received the B.Sc. degree in computer science from the Institute of Computer Science, Khwaja Fareed University of Engineering and Information Technology (KFUEIT), Rahim Yar Khan, Pakistan, in 2021, and the M.S. degree in computer science from KFUEIT. He has published several articles in reputed journals. His current research interests include data science, artificial intelligence, data mining, natural language processing, machine learning, deep learning, and image processing.

MUHAMMAD QASIM is currently pursuing the Bachelor of Science degree in cyber security from the Department of Cyber Security, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan, Pakistan. His current research interests include network security and artificial intelligence.

SARATH KUMAR POSA received the M.Sc. degree in information quality from the Department of Information Science, University of Arkansas at Little Rock, AR, USA, and the B.Tech. degree in electronics and communications engineering from Indian Institute of Information Technology D&M, Jabalpur, India. He has worked in different domains including finance, healthcare and retail. His current research interests include data science, artificial intelligence, data mining, machine learning, deep learning, and image processing.

ANCA DELIA JURCUT (Member, IEEE) received the B.Sc. degree in computer science and mathematics from the West University of Timişoara, Romania, in 2007, and the Ph.D. degree in security engineering from the University of Limerick (UL), Ireland, in 2013, funded by the Irish Research Council for Science Engineering and Technology. She was a Postdoctoral Researcher with UL, a member of the Data Communication Security Laboratory, and a Software Engineer with IBM, Dublin, Ireland, in the area of data security and formal verification. She has been an Assistant Professor with the School of Computer Science, University College Dublin (UCD), Ireland, since 2015. She has several key contributions to research focusing on the detection and prevention techniques of attacks over networks, the design and analysis of security protocols, automated techniques for formal verification, and security for mobile edge computing (MEC). Her research interests include security protocols design and analysis, automated techniques for formal verification, network security, attack detection and prevention techniques, security for the Internet of Things, and the applications of blockchain for security and privacy. For more information visit the link (https://people.ucd.ie/anca.jurcut).