Reinforcement Learning Framework for Multi-Class Intrusion Detection

1st Abhay Jogenipalli
Ira A. Fulton School of Engineering
Arizona State University
Tempe, AZ, USA
ajogenip@asu.edu

2nd Kumar Hasti
Ira A. Fulton School of Engineering
Arizona State University
Tempe, AZ, USA
khasti@asu.edu

3rd Kruthika Suresh
Ira A. Fulton School of Engineering
Arizona State University
Tempe, AZ, USA
ksures21@asu.edu

4th Rahul Tallam
Ira A. Fulton School of Engineering
Arizona State University
Tempe, AZ, USA
rtallam@asu.edu

5th Rushir Bhavsar (**Deputy Leader**)
Ira A. Fulton School of Engineering
Arizona State University
Tempe, AZ, USA
rbhavsa4@asu.edu

6th Sakshi Sheth
Ira A. Fulton School of Engineering
Arizona State University
Tempe, AZ, USA
ssheth26@asu.edu

7th Subharajit Pallob (**Leader**)
Ira A. Fulton School of Engineering
Arizona State University
Tempe, AZ, USA
spallob@asu.edu

8th Varshil Shah
Ira A. Fulton School of Engineering
Arizona State University
Tempe, AZ, USA
vshah50@asu.edu

***Abstract*—With increasingly sophisticated cyber threats, preserving network integrity and data security is critical for large-scale systems. Traditional measures like firewalls and access controls are often insufficient due to their vulnerability to exploitation. This project proposes enhancing intrusion detection systems (IDS) using machine learning (ML), deep learning (DL), and reinforcement learning (RL). We aim to evaluate multiple ML and DL models for accurate anomaly detection. Building on this, we plan to develop a novel two-step framework where an RL layer improves attack classification accuracy and reliability. This approach aspires to provide a more robust, real-time solution for detecting cyber threats.**

***Index Terms*—Cybersecurity, Intrusion Detection Systems, Machine Learning, Deep Learning, Reinforcement Learning, Anomaly Detection, Network Security, Data Protection, Attack Classification, Computational Learning Techniques**

I. Project Title and Focus

A Reinforcement Learning Framework for Multi-Class Intrusion Detection System Using Machine Learning and Deep Learning.This paper report presents a framework that uses reinforcement learning along with machine learning and deep learning to improve the accuracy and adaptability of multi-level intrusion detection systems for better threat detection in networks.

II. Problem Statement

Preserving network integrity and maintaining system-data security has been the number one priority for large-scale commercial systems. Subsequently, various methods have been designed to safeguard and protect network systems. The developed systems include repudiation, restriction-based access control, and system firewalls. However, such systems can be bypassed and avoided while creating an access path and exploiting system data; hence, a system needs to detect any incoming intrusions and make the system aware of any system-vulnerability exploitation. Systems employing ML techniques can detect such vulnerability attacks by analyzing various network traffic patterns and attributes. The major challenge faced in designing such a system is selecting the best computational learning technique and training such learning models to identify regular network traffic and network traffic anomalies and classify the attack type based on the pattern analyzed. This research article aims to establish a descriptive comparison to address the above mentioned difficulty.

III. Objectives

The objective of this paper is to evaluate the baseline performance of multiple machine learning models, create a Deep Learning Model and enhance it's effectiveness by incorporating state-of-the-art reinforcement learning techniques. This approach serves as a two-step verification system, where the reinforcement learning layer operates on top of the machine learning models to improve the accuracy and reliability of network attack classification. By leveraging both ML and RL methods, the paper aims to provide a more robust solution for detecting and classifying cyber threats.

IV. Background Research

A. Network Security, Cryptography and Intrusion Detection

Security has emerged as the most pressing requirement in our online world since privacy violations and data breaches occur almost as frequently as in the previous decade. An essential aspect of security is the unpredictability or randomness that cryptographic algorithms must include in the cipher or encrypted text sent to make the decryption process extremely difficult. The most crucial idea in cryptography for security entails various methods where the data is vulnerable to change or where an additional, highly secure random data point is put into the information. Homomorphic encryption is one of the essential cryptographic techniques out of all the ones available. The cipher-text is included in the transmission by performing cryptographic operations on the cipher text using the encryption method known as homomorphic encryption (HE). In the article [1], the authors present homomorphic cryptosystems to preserve security, characteristics, and homomorphic encryption categories. Furthermore, in [1], Patel et al. investigated the privacy-preserving applications of homomorphic cryptosystems in the cloud computing domain, retrieval of sensitive and private data, and information collation in a wireless sensor network. The study ([1]) goes into great depth on homomorphism and encryption approaches and their applications.

Many such algorithms in the purview of secure transmissions have security vulnerabilities where intruders or attackers can easily break down the cipher blocks to their basic form. Considering the type of files surging around on the World Wide Web, multimedia files such as images and videos require an extended form of encryption. The piping of such various algorithms can maintain a stronghold on the transmissions. To support and strengthen the above point of discussion, in [2], Oza et al. examined six different image encryption algorithms and proposed a new image encryption algorithm based on the Rubik's Cube Principle. Despite the algorithm's substantial computational strength for handling large multimedia files, it significantly strains system performance due to its high computational overhead. This leads to considerable processing times. Additionally, the potential loss of the encryption and decryption keys represents a critical vulnerability. These factors pose significant liabilities when the system needs to be scaled for large-scale projects. Lastly, the implementation of the algorithms across a common plane is presented as the future scope of the article, wherein the implementation presents the need for a simulator.

In [3], the authors focused on the flow correlation class of traffic analysis attacks in their study, wherein an attacker attempts to examine network traffic and correlate traffic from an input link with traffic from an outgoing connection. Further in their analysis, the authors have considered time-domain and frequency-domain correlation methods. Following the performance of extensive experimentation in the study of mixed networks based on their threat models, in [3], the authors found that except for a few batching schemes, all fail against flow-correlation attacks, permitting the attacker to identify entry or exit point locations of a flow or reconstruct the flow's trajectory.

In [4], the authors concentrate on developing an intrusion detection system on a Linux operating system and analyzing traffic, threats, and vulnerabilities with a configured firewall. Meanwhile, in [5], the researchers discussed intrusion detection systems and the issues and challenges faced in such systems, and the need for such systems. The authors [5] next examine the classification and model output optimization techniques frequently employed in IDS, noting their advantages and disadvantages while also presenting potential alternatives for future IDS research.

B. Machine Learning in Intrusion Detection and Network Security

Authors in [6] discuss six types of IDSes and further discuss and compare five algorithms for classification using the NSL-KDD data set. In [6], the authors compare five categories of attacks for the performance evaluation of classifier algorithms in the research; the attacks are DOS, R2L, U2R, Probe, and Routine. Out of the five algorithms—J48, Naive Bayes, Random Forest, Random Tree, and RepTree—the Random Forest algorithm gives the highest accuracy of correct instances at 99.9737%. Various algorithms work better for different types of attacks. The authors in [6] analyze and compare several strategies used in detection and intrusion based on the categorization of the attacks in the network. However, there are various attacks, so generalizing issue solutions is impossible.

The host-based intrusion detection system is one of the many IDSes that leverage system call patterns to better recognize normal and aberrant program behavior. In the article [7] to examine system call patterns and detect malicious patterns or intrusions, the authors propose merging Bag-of-Words (BoW) methodologies with AI-enabled machine learning techniques. The study's proposed system for pattern pre-processing employs the Bag-of-Words technique, with additional improvements such as Boolean value, probability value, and TF-IDF. The performance of classifier models is then evaluated using machine learning methods. J48 (C4.5), Random Forest, RIPPER, KNN, SVM, and Naive Bayes ML techniques are among those used. Finally, the performance of the proposed system was assessed based on detection precision and false alarm rate utilizing the ADFA-LD public access dataset and the authors' proposed virtual machine monitor (VMM) malware attack data collection in [7]. The Random Forest approach surpasses other ML algorithms in terms of intrusion detection accuracy and false alarm rate on the datasets used by the authors. When analyzed using ML algorithms, the study's proposed data set performs better than ADFA-LD.

Due to the prevalence of greater dimensional data and data correlation complexity, deep learning techniques have nearly doubled in the previous 5–6 years. Early identification of abnormalities and other qualities in security and secure systems is frequently identified or classified using DL approaches. In the face of continually developing network attacks, DL can

assist in the construction of an efficient IDS. However, DL approaches suffer from overfitting due to complicated network structures and high-dimensional data sets. Dropout and regularization are two adequately understood DL principles utilized for dealing with overfitting issues in order to improve the effectiveness of DL approaches.

To assess and enhance the effectiveness of Deep Neural Network-based IDS, the authors of the study [8] mix numerous regularization methodologies, such as L1, L2, and elastic net regularization further with a dropout regularization strategy. The datasets NSL-KDD, UNSW NB-15, and CIC-IDS-2017 are used in the research work by the authors [8]. The authors calculate the dropout probability using a hyperparameter optimization approach based on GridSearchCV. The paper uses a combination of regularization techniques to develop, analyze, and compare several cutting-edge machine learning algorithms for intrusion detection and classification, including Deep Neural Network (DL) techniques. The empirical analysis in the cited publication ([8]) demonstrates that dropout outperforms L1, L2, and elastic net regularization among the approaches utilized. Additionally, dropout, in combination with other regularization techniques, outperformed L1, L2, and elastic net regularization on their own.

In the research article [9], authors' Kathiresan et al. discuss various algorithms that can be used for intrusion detection and intrusion prevention. Further, [9] they discuss the best ML algorithms that can be used for intrusion detection—Fuzzy C-Means, Logistic Regression, KNN, Decision Tree, Random Forest, SVM, Naive Bayes, and Naive Bayes modification—and the article also discusses the pros and cons of using each of the algorithms. However, the paper does not show these algorithms' experimental or practical applicability.

Network security is and has become a significant necessity, wherein for the secure data communication or transmission of the data, detection of early intrusion is also needed for added security. Various factors and attributes contribute to a working network, and these attributes are strictly used as features in AI-based techniques of such detection or classification. Hence, more features lead to higher complexity, so reducing the dependency is vital for easier detection of intrusions. Reducing the features majorly affects the computational complexity but also profoundly impacts the algorithms used.

Therefore, the article [10] proposes a feature selection approach based on classification and regression trees (CART) that provides the best collection of features. The suggested approach in the article has also provided an optimal array of features passed across several classifiers for training and testing to develop a network intrusion detection system (NIDS). The authors in the study [10] investigated the performance accuracy of different existing machine learning (ML)-based classification methods and found that improved performance accuracy may be attained at a reduced computational cost. The suggested approach in the article has the lowest time complexity and highest accuracy in IDS design.

Yedukondalu et al. [11] discuss IDS and compare two ML algorithms—SVM and ANN. The authors of the article test to determine if the test data represents a regular user or a system assault. The study also covers typical discussions of SVM and ANN algorithms. The article's results showed that SVM performed quite severely, providing an accuracy of just 48%. ANN, on the other hand, provided an accuracy of 96%. The research also provided a framework for detecting intrusions, employing a GUI interface to get the dataset and carry out all operations necessary to predict intrusion attacks using the two algorithms.

In the article [12], Halimaa et al. discuss IDS, ML, and the algorithms employed in the task to predict the intrusion, wherein the article compares SVM and Naive Bayes algorithms for the task. Traditional SVM and Naive Bayes were utilized by [12] along with two variations for performance enhancement, namely CFSSubsetEval attribute selection and normalization. The accuracies presented by [12] are that SVM, along with the variations, has 97.29%, 93.95%, and 93.95% accuracy in the prediction task. Naive Bayes and the two variations have accuracies as 67.26%, 56.54%, and 71.00%. The misclassification rate is very high for Naive Bayes compared to that of SVM. It is clear from the results that SVM performs better against other algorithms.

In the research article [13], Mendoncca et al. concentrated on enhancing the IIoT's intelligent decision-making process, where an AI approach was put into practice by the authors in which the main cybersecurity assaults are susceptible to prediction using a deep learning model. The study's authors describe a unique prediction model that uses sparse evolutionary training (SET) to analyze and identify numerous security and integrity aspects. Precision, recall, accuracy, and F1-score are used as assessment metrics in the authors' thorough performance evaluation to examine the effectiveness of the proposed SET-based prediction model. The recommended SET-based model achieved an average accuracy of 0.99 with a testing duration of 2.29 ms. According to the authors, compared to existing state-of-the-art machine learning models, the proposed model improved attack detection accuracy by an average of 6.25% in a real-world scenario of IoT security in Industry 4.0. Given the complexity of the problem, the recommended unique strategy's accuracy is qualitatively reasonably high compared to baseline methods. Additionally, the accuracy achieved is good since the authors did the study for an open neural network, and the data is IoT-based. In contrast, the baseline approach produces a traditional high accuracy with confined memory ("fully-connected neural net layer" too).

In [14], authors Illavarason et al. address IDS and attack categorization using the KDD-99 dataset. [14] use three feature selection strategies to compare five ML algorithms. Correlation-based feature selection, Principal Component Analysis, and Information Gain Ratio-based feature selection are some techniques used. The algorithms include Naive Bayes, Support Vector Machine, Decision Tree, Neural Network, and K-Nearest Neighbour. PCA performs best in all three selection strategies, providing the maximum accuracy in all algorithms except the Decision Tree, which performs best in information gain and ratio-based feature selection.

Artificial intelligence (AI) and machine learning (ML) have become integral components in enhancing cybersecurity measures due to their ability to detect and respond to threats efficiently. In [15], the authors offer a comprehensive overview of the role of AI and ML in cybersecurity, discussing essential techniques, applications, challenges, and future directions. They review various ML algorithms for anomaly detection, malware classification, and network intrusion detection. The paper identifies several promising research opportunities for advancing AI-powered cybersecurity, including developing explainable AI, adversarial machine learning, transfer learning, autonomous and adaptive security, collaborative and federated learning, and integration with security orchestration and automation platforms. As cyber threats continue to grow in complexity, incorporating AI and ML into cybersecurity solutions will be essential for organizations to build more robust and adaptive defenses against evolving threats.

C. Reinforcement Learning and Machine Learning in Intrusion Detection and Cybersecurity

Moreover, reinforcement learning (RL), a subset of AI, has been applied to enhance intrusion detection systems. In [16], the authors present a novel multi-agent reinforcement learning architecture to enhance network intrusion detection systems through automation, efficiency, and robustness. They introduce an improved version of the Deep Q-Network (DQN) algorithm that incorporates a weighted mean square loss function and utilizes cost-sensitive learning techniques. Additionally, the model is designed to easily accommodate new attack patterns, ensuring its adaptability in evolving threat landscapes.

Similarly, [17] introduces MASFIDS, a reinforcement learning-based intrusion detection model for multi-agent feature selection networks. This model employs a feature self-selection algorithm within a multi-agent reinforcement learning framework, transforming the traditional 2^N feature selection space into N agent representations. This approach reduces model complexity and improves the search strategy for feature selection. Additionally, a Graph Convolutional Network (GCN) method is developed to extract deeper features from the data, ensuring accurate feature representation and expediting the selection process. To enhance accuracy further, mini-batches are utilized for data recording, allowing the integration of reinforcement learning with supervised learning techniques.

The development and training of autonomous cyber defense agents require high-fidelity environments that can simulate realistic scenarios. In [18], the authors introduce CyberWheel, a high-fidelity training environment that combines simulation and emulation capabilities for agent development. CyberWheel facilitates the customization of training networks by allowing users to easily redefine the agent's reward function, observation space, and action space, promoting rapid experimentation with novel agent designs. Additionally, it offers visibility into agent behaviors, which is crucial for evaluation, and provides comprehensive documentation and examples to lower the user entry barrier.

Overall, the integration of AI, ML, and RL into cybersecurity frameworks has shown significant promise in enhancing threat detection and response capabilities. The incorporation of advanced techniques such as multi-agent reinforcement learning and feature self-selection algorithms addresses the challenges posed by evolving cyber threats. As research progresses, the development of adaptive and autonomous systems will be essential in maintaining robust cybersecurity defenses.

V. METHODOLOGY

This paper presents a robust and scalable framework that integrates machine learning (ML), reinforcement learning (RL), and transfer learning techniques to enhance multi-class intrusion detection. Utilizing the NF-UQ-IDS-V2 dataset from Kaggle as a reliable foundation, the methodology employs comprehensive data preprocessing and feature engineering for optimal model development. The proposed intrusion detection pipeline combines transfer learning, majority-vote classification, and a feedback loop to ensure high performance. This highly adaptive approach is capable of detecting a wide range of network-based attacks in real time, resulting in a dynamic intrusion detection system with high detection and classification accuracy and adaptability to emerging threats.

A. System Model

The system model encompasses several stages to create a comprehensive intrusion detection pipeline. Figure 1 illustrates the contiguous flow of techniques and algorithms in this framework.The process begins with the NF-UQ-IDS-V2 dataset from Kaggle, a widely recognized resource for network intrusion detection tasks that provides a robust foundation for model development.

In the first stage, rigorous data preprocessing is conducted, including handling missing values, normalizing features, and transforming the dataset for modeling. This step is critical for standardizing input features, ensuring all data points are on a comparable scale, reducing biases, and improving model convergence. Following preprocessing, feature engineering extracts the most relevant information from the dataset. Techniques such as feature-to-target correlation analysis, mutual information, and recursive feature elimination (RFE) are employed to identify the most informative features. This reduces data dimensionality and optimizes model performance.

Once refined, the dataset is benchmarked using a suite of machine learning (ML) algorithms and, in parallel, a deep learning fully connected neural network (DL-FCNN) model. After benchmarking, transfer learning is applied by using the pre-trained deep learning model as the base for a reinforcement learning-based Deep Q-Network (RL-DQN). This enables the system to leverage prior knowledge from the deep learning model, accelerating the RL-DQN's training process. The RL-DQN interacts with network traffic, learning through a reward-based system where it takes actions—such as identifying attacks—and refines its decision-making policy to maximize prediction accuracy. Its ability to learn dynamically

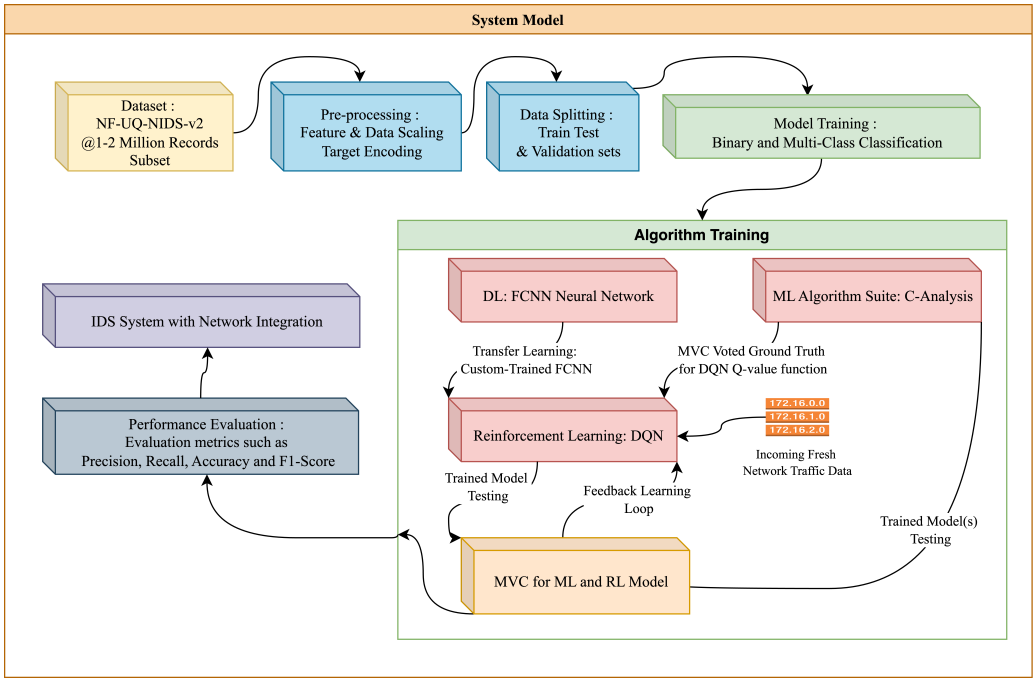

Fig. 1. AI based IDS Framework: Reinforcement Learning Techniques + ML/DL Algorithms

from the environment makes it adaptable to evolving network conditions and attack patterns.

In parallel, a Majority-Voting Classifier (MVC) is incorporated to enhance classification robustness. The MVC combines outputs from both the ML algorithms and the RL-DQN model in an ensemble-based approach, leveraging their strengths to provide more accurate and resilient classification, especially in multi-class scenarios with varied and complex attacks. A crucial innovation in the proposed framework is the feedback loop, which allows continuous learning and improvement. The MVC's output is fed back into the RL-DQN model, forming an iterative learning process that refines the RL model's detection capabilities over time. This feedback mechanism ensures the system remains adaptive, improving performance as new types of attacks emerge or network behaviors evolve. The iterative nature of the feedback loop creates a self-improving system capable of adjusting to real-time data and detecting sophisticated attacks more accurately.

B. Dataset

To perform an in-depth analysis of network intrusion detection and attack classification, we selected the NF-UQ-NIDS-v2 dataset [19], accessible from Kaggle.

This comprehensive dataset was developed and compiled by the University of Queensland, specifically designed for network intrusion detection system (NIDS) research and openly available for public use. The NF-UQ-NIDS-v2 dataset offers a comprehensive collection of network traffic records, encompassing both benign activities and a wide array of malicious attacks. Table I tabulates the details of the usage of the dataset procured for the experiment and findings presented in this paper.

TABLE I
NF-UQ-NIDS-v2 DATASET INFORMATION

Source	*University of Queensland*
Total attributes	*43*
Records	*11,994,893*
Records used	*450,000*
Total Target label Classes	*2-Normal and Attack, 21 Attack-types*

The diversity of attack types available in the dataset, makes it crucial for training and developing a reliable and generalizable Intrusion Detection System (IDS) model, capable of distinguishing between benign and malicious network behavior across various configurations. Given the evolving complexity of cyber threats, the NF-UQ-NIDS-v2 dataset is particularly well-suited for the machine learning (ML), deep learning (DL), and reinforcement learning (RL) approaches employed in this study. Its rich set of attributes and extensive record count enable effective anomaly detection and precise classification, which are essential for our proposed two-step framework. Table II categorically lists the attack types available in the Dataset.

C. Data Preprocessing

The NF-UQ-NIDS-v2 dataset includes a diverse range of network traffic metrics, from small integers to large values representing data packet sizes in bytes. To optimize these values for accurate classification, we applied a series of

TABLE II
DATASET ATTACK TYPES INFORMATION

Attack-Type	Attack Sub-type
dos *attacks*	Hulk, *slowhttptest*, *goldeneye* and *Slowloris*
DDOS *attacks*	LOIC-UDP, HOIC and LOIC-HTTP
bruteforce *attacks*	FTP-bruteforce, SSH-Bruteforce, Brute Force-Web and Brute Force-XSS
Others	Analysis, Backdoor, Benign, Bot, Exploits, Fuzzers, Generic, Infiltration, Reconnaissance, Shellcode, Theft, Worms, injection, MITM, password, ransomware, scanning.

preprocessing techniques to normalize data, handle categorical variables, and select key features. After verifying the dataset's integrity by checking for duplicates and missing values, we applied Min-Max Normalization to scale numerical features within a 0-1 range, which enhances model performance by standardizing feature ranges. This normalization is especially beneficial in intrusion detection tasks, as highlighted by authors in [20], who demonstrated that proper preprocessing, including normalization and categorical handling, significantly boosts model accuracy and minimizes false positives. Similarly, authors in [21] emphasized that techniques like Min-Max scaling are essential for improving convergence rates and detection performance in machine learning models for NIDS applications. Together, these steps ensure our dataset is optimally prepared for our framework.

To handle categorical attributes effectively, One-Hot Encoding (OHE) was applied, particularly for the target column and protocol-related features, transforming categorical information into a numerical format suitable for machine learning and deep learning algorithms. IP addresses were encoded in a way that preserves the underlying network structure, avoiding bias from specific network identifiers. For dimensionality reduction, Recursive Feature Elimination (RFE) was employed, identifying the 21 most influential features for network anomaly detection and attack classification.

Studies underscore RFE's advantages in feature selection. For instance, authors in [22] demonstrated that RFE, through Fibonacci and k-Subsecting approaches, efficiently identifies smaller feature subsets without sacrificing predictive performance, compared to conventional selection methods. Similarly, authors of the paper [23] showed that Conformal Recursive Feature Elimination (CRFE) enhances model robustness by strategically removing features that increase non-conformity, resulting in a more refined feature set. These findings validate RFE's strategic advantage in balancing efficiency with performance, as it narrows the dataset to essential features, enhancing computational efficiency without compromising classification accuracy. The selected features effectively support both binary classification (normal vs. anomaly) and multi-class classification tasks, enabling robust applications of ML and DL methods as detailed in the following sections.

D. Feature Selection

To optimize model accuracy and reduce computation time, we performed feature selection on the NF-UQ-NIDS-v2 dataset, utilizing Recursive Feature Elimination (RFE) with a Random Forest Classifier as an external estimator. RFE identified 21 significant features essential for intrusion detection and classification tasks, ensuring efficient model performance without loss of accuracy.

The subset of the selected features, such as FLOW_DURATION_MILLISECONDS, MIN_IP_PKT_LEN, and L7_PROTO, are outlined in Table III. The black-box nature of attribute choice via RFE cannot be explained in simple words. However the intuition behind the choice of features can be such, FLOW_DURATION_MILLISECONDS measures the total time span of a network flow; shorter durations are often indicative of scanning or denial-of-service attacks, where connections are brief and numerous. MIN_IP_PKT_LEN represents the smallest packet size within a flow, which is valuable in anomaly detection, as attack flows, such as reconnaissance or flooding, tend to include smaller packet sizes that differ from typical traffic patterns.

L7_PROTO identifies Layer 7 protocols, allowing the model to detect inconsistencies or misuse of application-layer protocols—often a sign of protocol exploitation attempts. Other features, like IPV4_SRC_ADDR and CLIENT_TCP_FLAGS, contribute by mapping source-specific behaviors and analyzing TCP flag patterns, which can reveal abnormal connection setups or unexpected flag sequences commonly associated with malicious intents. These features collectively form a focused dataset that is particularly adept at capturing key network dynamics, enabling effective anomaly detection and accurate classification of diverse attack types.

TABLE III
IMPORTANT FEATURES OF THE NF-UQ-NIDS-V2 DATASET

Important Feature
FLOW_DURATION_MILLISECONDS, MIN_IP_PKT_LEN, DURATION_IN, L7_PROTO, IPV4_DST_ADDR, DNS_QUERY_ID, SERVER_TCP_FLAGS, L4_SRC_PORT, LONGEST_FLOW_PKT, MAX_IP_PKT_LEN, CLIENT_TCP_FLAGS, TCP_FLAGS, TCP_WIN_MAX_OUT, IPV4_SRC_ADDR

This streamlined feature set effectively supports our machine learning and deep learning models, providing a focused dataset for accurate intrusion detection and classification.

VI. EXPERIMENT SETUP

The experimental setup for this study involved conducting all analyses on Kaggle's online notebook platform using a P100 GPU to optimize processing power. The experiments were designed around three core approaches—Machine Learning (ML), Deep Learning (DL), and Reinforcement Learning (RL)—each contributing to the development of an adaptive and scalable intrusion detection model.

To maintain consistency across these approaches, data preparation steps, including data cleaning, normalization,

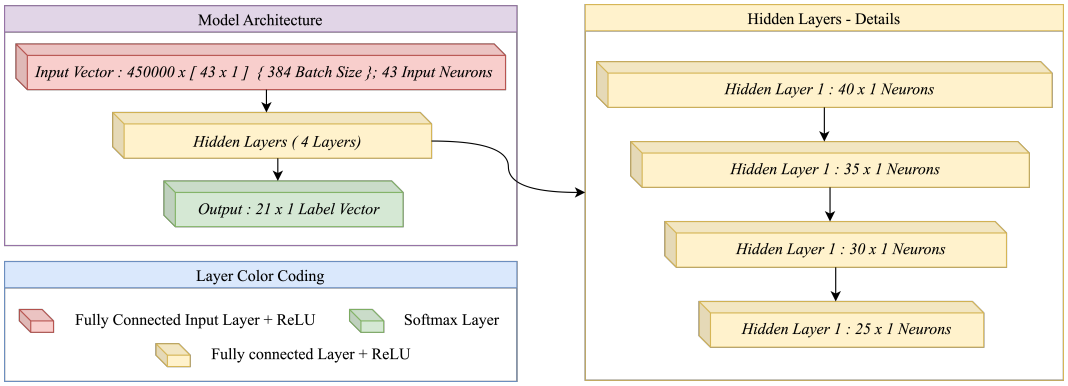

Fig. 2. Architecture of DL Neural Network

and feature selection through Recursive Feature Elimination (RFE), were applied uniformly. This streamlined dataset, containing 21 key features, provided a strong foundation for all model types.

In the ML approach, nine standard algorithms were trained to establish a comparative performance baseline. The DL approach then implemented a Dense Neural Network (DNN) to benchmark deep learning performance. Extending from this baseline, an RL-enhanced Deep Q-Network (RL-DQN) model was developed, incorporating reward-based learning to dynamically improve classification accuracy over time.

To consolidate the strengths of each model, a Majority Voting Classifier (MVC) ensemble was applied, integrating outputs from the ML, DL, and RL models for more robust and accurate predictions. Evaluation metrics, including accuracy, precision, recall, and F1 score, were used to assess model performance, ensuring a reliable foundation for intrusion detection in complex network environments.

A. *ML for Anomaly Detection and Attack Classification*

In this approach, two experiments were performed; firstly, a cross-validation scheme wherein the performance score was validated for five training set splits. Secondly, a direct fit and predict strategy was followed to train all the classifiers and predict the anomaly and attack-type detection.

*1) **Algorithms Selection**:* The selection of these nine algorithms was strategically designed to cover a spectrum of complexity, prediction accuracy, and adaptability to diverse dataset characteristics, enhancing both the detection of anomalies and the classification of attack types.

- Simpler algorithms like ***Logistic Regression***, ***Bernoulli Naive Bayes*** [24], and ***Gaussian Naive Bayes*** [25] were chosen for their computational efficiency and capacity for fast classification. These algorithms operate effectively with assumptions of feature independence and provide a straightforward probabilistic interpretation, making them ideal for rapid evaluation and baseline comparisons.
- ***Support Vector Machine (SVM)*** [26] was selected for its strength in handling high-dimensional datasets. Unlike simpler models, SVM effectively manages feature correlation and complex decision boundaries, making it highly suitable for datasets with intertwined attributes—a common characteristic in network traffic data.
- Tree-based algorithms, including ***Decision Trees*** and ***Random Forest*** [27], offer a robust alternative by constructing models that capture hierarchical relationships within data, thus providing resilience to variance across samples. Random Forest, as an ensemble method, minimizes overfitting by aggregating multiple decision trees, leading to improved generalization and predictive accuracy, particularly when data exhibits heterogeneity or inconsistencies.
- ***Gradient Boosting***, ***LightGBM***, and ***XGBoost*** are incorporated for their advanced boosting techniques, which iteratively refine model predictions by focusing on errors from previous iterations [28]. These algorithms excel in classification precision by minimizing bias and variance, which is essential for accurately detecting nuanced attack patterns. XGBoost, in particular, is highly optimized for speed and performance, making it well-suited for large-scale anomaly detection tasks.

Together, these selected algorithms provide a balanced framework for building a robust and accurate intrusion detection system, leveraging simplicity where appropriate and complexity where needed for precision.

*2) **Model Setup**:* Direct baseline classifiers were used after studying and pre-processing the data, with 80-20 training-testing split, wherein no explicit hyper-parameter tuning was carried out, considering the task at hand of classification.

B. *DL for Anomaly Detection and Attack Classification*

*1) **Architecture Selection and Design**:* The ***Fully Connected Dense Neural Network (DNN)*** architecture was selected for intrusion detection due to its effectiveness in capturing complex, non-linear relationships within network traffic data. This architecture, depicted in Figure 2, begins with an input layer handling 43 features and progresses through four hidden layers with decreasing neuron counts $(40, 35, 30,$ *and* $25)$, ending with a binary output layer.

This layered design enables hierarchical feature abstraction, which is crucial for identifying subtle patterns associated with

intrusions. Authors in the study [29] have shown that DNNs, even in relatively simple configurations, perform well in anomaly detection by learning layered feature representations that enhance classification accuracy.

Additionally, in a related research [30], researchers demonstrates that DNNs are well-suited for dynamic environments, as they adapt effectively to variations in network behavior, which is essential for real-time intrusion detection. By balancing model depth with computational efficiency, this architecture provides a robust yet scalable solution for accurately identifying attack types in network intrusion detection tasks.

*2) **Hyper-parameters**:* For the employed DL architecture of a Dense neural network, the hyper-parameters are tabulated in Table IV, including 300 epochs for sufficient training and a batch size of 384 to optimize memory use. Categorical Cross-entropy is used for multi-class classification, with 'adam' as the optimizer for efficient convergence. Accuracy is the primary performance metric, while 'relu' and 'softmax' activation functions in hidden and output layers, respectively, ensure effective non-linearity and probability-based outputs. These settings are tuned for optimal classification performance.

TABLE IV
NEURAL NETWORK HYPER-PARAMETERS

Hyper-Parameter	Value
Epochs	300
Epoch Batch Size	384
Loss Function	Categorical Cross-entropy
Model Optimizer	adam
Performance Metric	Accuracy Measure
Activation Function	'relu', 'softmax'

C. Deep Reinforcement Learning Approach for Intrusion Detection and Attack Classification

*1) **Architecture Selection and Design**:* In designing the architecture for our multi-class intrusion detection system, we employed a Deep Q-Network (DQN), a reinforcement learning algorithm that integrates Q-learning with deep neural networks to effectively handle complex and expansive state spaces. The DQN's architecture is initialized using transfer learning from a pre-trained deep learning model, enabling the system to leverage existing knowledge of network traffic patterns.

This initialization facilitates the DQN's ability to refine its classification decisions through a reward-based learning process, enhancing its adaptability to evolving network behaviors. Authors of the research [31] demonstrated the efficacy of DQN architectures in network intrusion detection, highlighting their capacity to learn and adapt to dynamic environments.

Additionally, authors of the work [32] underscores the scalability of DQNs in managing large-scale data, which is crucial for processing extensive network traffic. By integrating transfer learning with a DQN framework, our system achieves a robust and scalable solution for accurate and efficient intrusion detection.

TABLE V
DQN MODEL HYPER-PARAMETERS

Hyper-Parameter	Value
Learning Rate (α)	0.001
Discount Factor (γ)	0.99
Exploration Rate (ϵ)	Decayed from 1 to 0.01
Batch Size	64
Timesteps	500,000

*2) **Hyper-parameters**:* The Deep Q-Network (DQN) performance hinges on optimized hyper-parameter selection, as shown in in Table V. A learning rate (α) of 0.001 ensures stable, gradual updates, while a high discount factor ($\gamma = 0.99$) captures long-term dependencies in network traffic patterns.

The exploration rate (ϵ) decay from 1 to 0.01 allows the DQN to balance exploration and exploitation, aligning with strategies outlined by Alavizadeh et al. (2021) [31]. A batch size of 64 and Episode Timestep of 500,000 provide adequate convergence without overuse of computational resources. These settings are critical to achieving robust, accurate classification in intrusion detection.

*3) **Model Setup**:*

*a) **Input (States):**:* Each state is defined by a set of features extracted from the NF-UQ-IDS-V2 dataset, representing individual network traffic samples. The selection of these features is crucial, as they must capture all relevant information needed for the DQN to distinguish between normal traffic and various attack types.

*b) **Output (Actions):**:* The action space consists of 21 discrete actions corresponding to specific classification labels. Actions in the DQN model represent specific classification labels, subset of which is detailed in Table VI.
To note, this approach in experimentation incorporates actions in the training and feedback loop to agent for computation of the rewards, wherein no actual action is taken in the environment.

TABLE VI
ACTION SPACE AND CLASSIFICATION LABELS

Action	Classification
0	Classify as "Normal"
1	Classify as "DoS" (Denial of Service Attack)
2	Classify as "DDoS" (Distributed Denial of Service Attack)
3	Classify as "Brute Force" (Password guessing attacks)
4	Classify as "SQL Injection" (Database command injection attacks)
5	Classify as "Botnet" (Malware-controlled network traffic)

*c) **Reward Structure:**:* The reward function is a pivotal hyper-parameter that guides the learning process by providing

TABLE VII
MODEL RESPONSES AND LOGGING FOR EACH ACTION

Action	Response	Alert and Logging
0: Normal Traffic	Allow network traffic through, as it is classified as normal and does not pose a threat.	Minimal or no logging, as this traffic is benign.
1: DoS Attack	Block or rate-limit the traffic to prevent overwhelming the network.	Trigger an alert to the network administrator, log the IP address, and record details like timestamps and traffic volume for further analysis.
2: DDoS Attack	Block the IP addresses or throttle traffic volume from sources identified as part of the DDoS attack.	Generate a high-priority alert and log comprehensive details. If possible, initiate automated responses, such as reconfiguring firewalls to temporarily block traffic from specific sources.
3: Brute Force Attack	Block the IP address or user attempting access or apply temporary access restrictions to slow down repeated login attempts.	Log login attempts, IP addresses, and alert the administrator to check for unauthorized access. Record details such as the number of failed attempts, timestamp, and targeted resource.
4: SQL Injection Attack	Immediately block the specific query or request associated with the SQL injection attempt.	Log request details, including the source IP, timestamp, and SQL query details if possible. Generate a high-priority alert to inform database and security teams.
5: Botnet Activity	Block or quarantine traffic from devices showing botnet-like behavior.	Send an alert to network administrators and log device details. Consider initiating a broader network scan to identify additional potentially infected devices.

feedback on the agent's actions. The rewards are assigned to encourage correct classifications and discourage misclassifications, as outlined in Table VIII.

TABLE VIII
REWARD STRUCTURE FOR CLASSIFICATION ACTIONS

Reward	Reasoning
Correct Classification	
+2 Points	Encourages correct classification of network traffic.
Incorrect Classification (Attack as Normal)	
-3 Points	Heavy penalty to avoid missing attacks.
Incorrect Classification (Attack as Different Type)	
-1 Point	Less severe penalty than misclassifying as normal.

*4) **Feedback Loop and Retraining**:* The feedback loop in this DQN model actively tunes hyper-parameters based on misclassification patterns, ensuring the model continuously adapts to complex network behaviors. Misclassified instances are analyzed temporally, with the reward function dynamically adjusted to penalize recurring misclassifications, thereby incentivizing the model to prioritize learning these challenging patterns, while refining the model's Q-values. Additionally, the feedback loop leverages performance metrics, such as precision and recall, to adjust key hyper-parameters like the learning rate, episode count, and replay buffer size.

For instance, if recall is low for specific attack types, the learning rate may be increased, or the replay buffer expanded to incorporate more diverse instances, enabling deeper reinforcement learning from under-represented or complex patterns. This iterative process ensures that the model not only improves in accuracy over time but also maintains adaptability to new or emerging attack scenarios, achieving an optimal balance of learning speed and classification reliability.

*5) **Expected Model Responses for Each Action**:* A subset of the model's responses for each action, along with the corresponding alerts and logging procedures, are detailed in Table VII.This response structure allows the model to act logically based on the classification, enhancing security by providing appropriate responses for each detected traffic type.

VII. RESULTS: PERFORMANCE EVALUATION

A. Evaluation Metrics

The performance measures used in this work are calculated using a confusion matrix that indicates the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) in each category. These measures include:

- **Accuracy**
- **Precision**: Can adequately quantify the performance of models when the cost of false positives in prediction models is high.
- **Recall**: Concerned with circumstances in which actual faulty classes are projected to be non-defective.
- **F1-score**: Used in prediction classification tasks due to the imbalance of datasets.

B. Performance Evaluation: Machine Learning Algorithm Suite

Table IX presents the precision and recall metrics along with the average accuracy for the cross-validation training scheme across various ML classifiers, tested on the NF-UQ-NIDS-v2 dataset to assess their effectiveness in identifying network anomalies and intrusions.

Figure 3 provides a comparison of accuracy scores across ML classifiers in the NF-UQ-NIDS-v2 dataset for the multi-class attack classification task, emphasizing their performance in both intrusion detection and attack-type classification tasks.

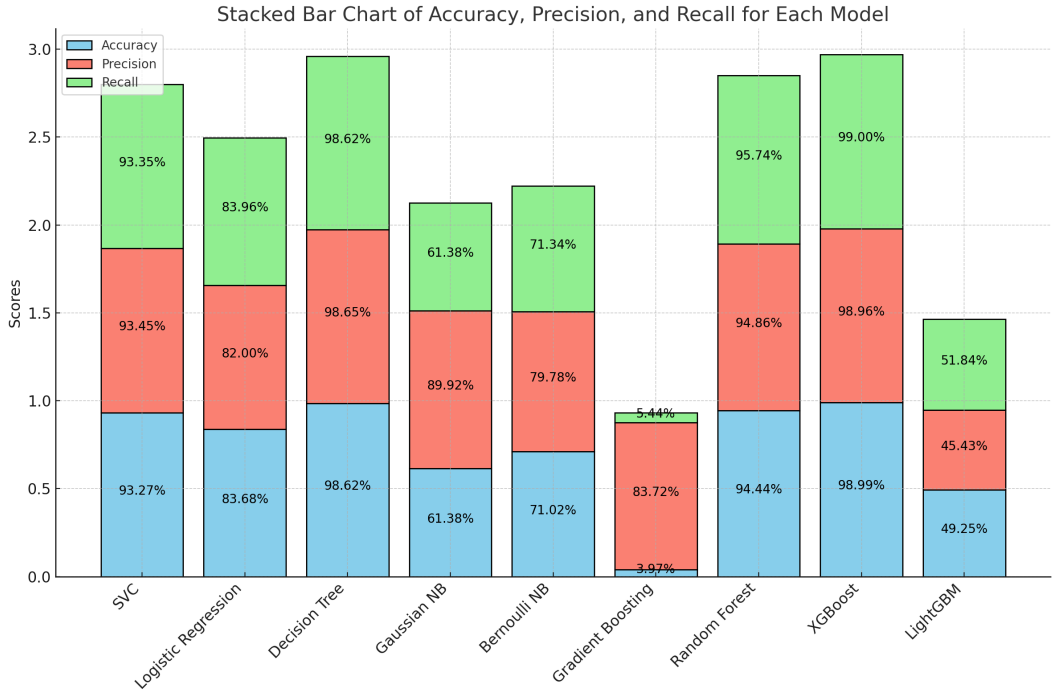

Fig. 3. Evaluation Measures for Multi-class Attack Classification for NF-UQ-NIDS-v2 Dataset

TABLE IX
EVALUATION MEASURES FOR BINARY CLASSIFICATION CROSS-VALIDATION IN NF-UQ-NIDS-V2 DATASET

ML Classifiers	**Precision (%)**	**Recall (%)**	**Accuracy (%)**
SVM	94.78 ± 0.17	97.96 ± 0.08	95.02 ± 0.07
LR Classifier	92.43 ± 0.26	88.27 ± 0.45	87.28 ± 0.32
Decision Tree	99.59 ± 0.0	99.62 ± 0.04	99.46 ± 0.03
Gaussian NB	87.91 ± 0.35	96.93 ± 0.35	88.99 ± 0.09
Bernoulli NB	95.77 ± 0.08	79.76 ± 0.09	84.06 ± 0.11
Gradient Boost	99.52 ± 0.03	99.52 ± 0.07	99.35 ± 0.04
Random Forest	99.25 ± 0.05	99.12 ± 0.08	98.92 ± 0.03
XGBoost	$\mathbf{99.7 \pm 0.04}$	$\mathbf{99.73 \pm 0.02}$	$\mathbf{99.61 \pm 0.02}$
LightGBM	99.68 ± 0.03	99.7 ± 0.03	99.58 ± 0.03

TABLE X
CLASSIFICATION REPORT AND BASELINE ACCURACY MEASURE FOR ATTACK-TYPE DETECTION USING DNN MODEL

Measure	**Precision (%)**	**Recall (%)**	**Accuracy (%)**
Training	96.64 ± 1.60	96.18 ± 2.44	96.64 ± 1.60
Validation	97.19 ± 0.81	97.03 ± 0.64	96.46 ± 1.22

C. Performance Evaluation: Deep Learning for Transfer Learning

Table X tabulates the Classification report and Training & validation accuracy for DL Model for Attack-type Classification approach towards Intrusion Detection and transfer learning based integration in RL-DQN Model.

Figure 4 shows a comparison graph for Training and Validation values of accuracy and Loss for every epoch of backprop, i.e., training the model.

D. Performance Evaluation: Proposed integrated Reinforcement Learning

Table XI tabulates the Classification report and Baseline accuracy Measure for a subset of 5 out of 20 attack-types in the Anomaly Detection-Classification Proposed Model, used as the integrated approach towards Intrusion Detection. The results have been displayed to a limit of 5 attacks as the scores remain consistent along the range of 0.96 0.99 for the remaining attack types.

TABLE XI
CLASSIFICATION REPORT AND BASELINE ACCURACY MEASURE FOR ATTACK-TYPE DETECTION FOR RL-DQN MODEL

Class	**Precision**	**Recall**	**F1-Score**
Normal (BENIGN)	1	0.99	0.99
DoS	1	0.99	0.99
DDoS	1	1	1
Brute Force	0.73	0.64	0.68
SQL Injection	0.01	0.65	0.03
Botnet (Bot)	0.35	0.99	0.51
Weighted Average	0.99	0.99	0.99

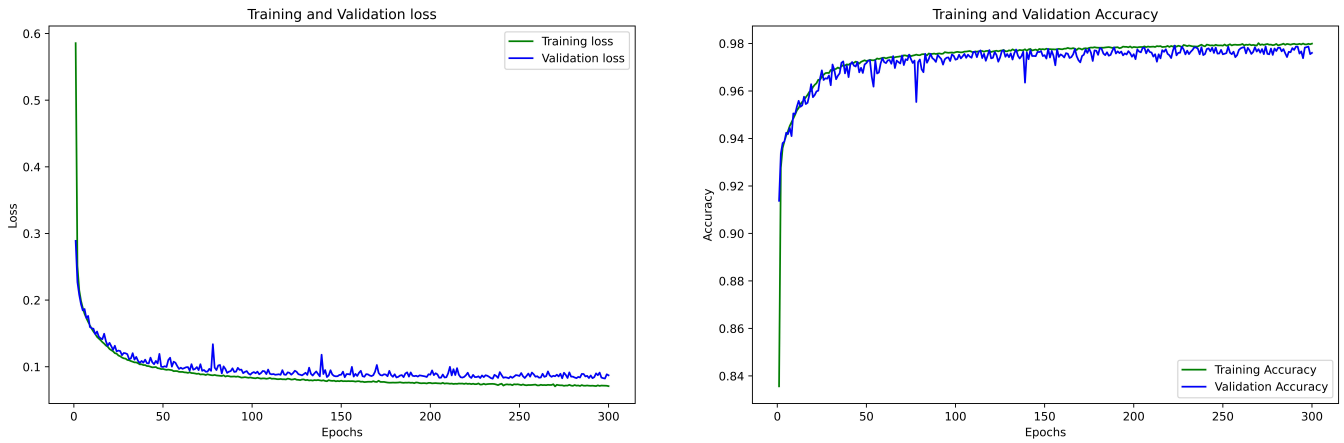

Fig. 4. Training and Validation Loss and accuracy Comparison

E. MVC Classifier Evaluation Measures

Baseline Accuracy for the combination of the ML and RL Model under Majority Voting Classification with equal weighted-basis ensembling was 99.82%.

VIII. Results Discussion

The core challenge outlined in the problem statement was the need for a robust system capable of accurately detecting network intrusions and classifying various attack types, despite complexities such as imbalanced datasets and the diverse nature of cyber threats. The objective was to evaluate multiple machine learning (ML) models and enhance their effectiveness by incorporating state-of-the-art reinforcement learning (RL) techniques, ultimately creating a two-step verification system that improves accuracy and reliability in network attack classification.

In the initial phase, we applied 9 ML algorithms to the NF-UQ-NIDS-v2 dataset to establish baseline performance metrics for the Binary Anomaly Detection task and the Multi-class Attack Classification Task. The algorithms include Support Vector Machines (SVM), Logistic Regression, Decision Trees, Random Forests, Gradient Boosting, XGBoost, and LightGBM. The performance of these models in binary classification (distinguishing between normal and anomalous traffic) was notably high. For instance:

- **Gradient Boosting**: Achieved equal precision and recall values at 99.52%, indicating strong performance in detecting anomalies.
- **SVM**: Recorded a low precision of 94.78% and recall of 97.96%, showing good balance between true positive and false positive rates.
- **Decision Trees**: Attained precision and recall both around 99.59% and 99.62% respectively, reflecting high accuracy in classification.
- **XGBoost**: Led the group with a precision of 99.7% and recall of 99.73%, highlighting its effectiveness in binary intrusion detection.

Despite these high evaluation measures in binary classification, the models' performance significantly remained consistent when extended to multi-class attack-type classification. The imbalanced nature of the dataset—where certain attack types were underrepresented—posed a challenge for these algorithms, considering the epistemic uncertainty in Network Traffic. For example, Naive Bayes classifiers struggled the most, with Gaussian and Bernoulli variants achieving only 61.38% and 71.02% accuracy, respectively, in attack-type classification. Gradient Boosting suffered the most, with accuracy as low as 3.97%. Advanced models like Random Forests and XGBoost showed success with Δ of 4.81% and 0.73% for the accuracy.

Recognizing the epistemic uncertainty of Network Traffic Attack Types in the data and inherent training limitations, we introduced a Deep Learning (DL) model as an intermediary step to enhance feature representation for the attributes to handle the imbalancing. The DL model, a fully connected Dense Neural Network, aimed to capture complex patterns in network traffic data. While the DL model achieved a weighted average precision and recall of 98%, its macro-average precision and recall were only 75% and 67%, respectively. This discrepancy indicates that while the model performed well overall, it was less effective in accurately classifying minority classes—reflecting a reduction in accuracy for underrepresented attack types. The black-box nature of the DL model made it challenging to interpret which features were most influential, potentially limiting its effectiveness in addressing the imbalanced dataset issue directly.

However, the DL model played a crucial role in enhancing the subsequent Reinforcement Learning (RL) phase. By providing richer feature representations, it supplied the RL model with a more informative state space to learn from. The Deep Q-Network (DQN) leveraged these representations and incorporated a feedback learning loop to iteratively improve its performance. The feedback loop involved reintroducing

misclassified instances back into the training set, allowing the model to focus on learning the characteristics of minority and less frequent attack types.

The RL-DQN model demonstrated significant improvements in multi-class attack-type classification. It achieved high precision, recall, and f1-score values across most attack classes. Notably, the model excelled in detecting challenging attack types that traditional ML models struggled with:

- **DDoS Attacks**: Achieved perfect precision and recall, indicating flawless detection.
- **Botnet Activity**: Improved detection rates with a precision of 35% and recall of 99%, a substantial enhancement over previous models.
- **SQL Injection Attacks**: Despite the inherent difficulty, the model achieved a recall of 65%, highlighting its ability to detect even the most underrepresented attack types.

The integration of the RL-DQN model within a Majority Voting scheme marginally increased the overall accuracy of the integrated approach. By combining the strengths of the ML models in binary classification and the RL model's enhanced multi-class capabilities, the ensemble achieved a baseline accuracy of 99.82%. This significant improvement demonstrates the efficacy of the proposed two-step verification system in meeting the research objectives.

In essence, the proposed methodology effectively addressed the challenges outlined in the problem statement:

1) **Handling Imbalanced Datasets**: The integrated approach of RL-DQN model allowed for focused learning on underrepresented classes, mitigating the impact of dataset imbalance and epistemic learning uncertainty.
2) **Improving Attack-Type Classification**: The enhanced feature representations from the DL model provided a robust foundation for the RL model to accurately classify a diverse range of attack types.
3) **Achieving High Accuracy and Reliability**: The combined ML-RL approach, validated through improved precision and recall metrics, confirmed the system's effectiveness in accurately detecting and classifying network intrusions.

By integrating ML, DL, and RL techniques, the research successfully developed a robust intrusion detection system that not only identifies anomalies but also accurately classifies the type of attack. This holistic approach leverages the strengths of each method: the interpretability and initial accuracy of ML models, the deep feature extraction capabilities of DL models, and the adaptive learning and decision-making proficiency of RL models. The results affirm that the objectives were achieved, providing a practical and effective solution for enhancing network security against a wide array of cyber threats.

IX. Conclusion

The proposed Reinforcement Learning approach, utilizing a Deep Q-Network (DQN) with a feedback learning loop and integrated into a Majority Voting Classifier (MVC), has proven to be robust and effective for intrusion detection and attack-type classification. The results demonstrate the approach's high accuracy and adaptability, significantly outperforming traditional machine learning methods in handling imbalanced datasets and complex multi-class scenarios.

By effectively learning from misclassifications through the feedback loop, the model continuously enhances its detection capabilities, particularly for rare or underrepresented attack types. The integration with the MVC further improves performance, resulting in a highly accurate and reliable intrusion detection system.

Given its demonstrated efficacy, this approach shows great promise for real-world deployment in network security environments. Its ability to accurately detect and classify a wide range of network attacks suggests that it would perform exceptionally well in practical applications, contributing significantly to the advancement of cybersecurity measures.

X. Future Work

Future work will concentrate on advancing the Reinforcement Learning (RL) capabilities of the DQN model to enhance its performance in intrusion detection and automated response. One potential direction is to incorporate tailored RL techniques involving Zero-day Attack avoidance architectures, which can mitigate issues like parellel-estimation of intrusions and improve learning stability. Exploring the use of prioritized experience replay for network attributes could also make the learning process more efficient by focusing on dangerous attacks.

Another significant avenue is to enable the RL agent to interact directly with the network environment. Integrating the model with network analysis tools and system interfaces would allow it to receive real-time state information and execute actions it has been trained on, such as adjusting firewall settings, terminating malicious processes, or isolating affected network segments. Ensuring safe exploration and adherence to security protocols will be essential to prevent unintended side effects. These enhancements aim to develop an autonomous and adaptive intrusion detection system capable of learning from and responding to evolving cyber threats in real-world network settings.

Appendix

A. Code

The code employed in this paper is resource heavy. For easy reference and testing, the *.ipynb* has been hosted on Google Colab for instance access. Link to the code: Google Colab.

REFERENCES

[1] N. Patel, P. Oza, and S. Agrawal, "Homomorphic cryptography and its applications in various domains," in *International Conference on Innovative Computing and Communications*, pp. 269–278, Springer, 2019.

[2] P. Oza, V. Kathrecha, and P. Malvi, "Encryption algorithm using rubik's cube principle for secure transmission of multimedia files," in *Third International Conference on Multidisciplinary Research and Practice IJRSI*, vol. 4, pp. 239–243, 2016.

[3] M. A. Qadeer, A. Iqbal, M. Zahid, and M. R. Siddiqui, "Network traffic analysis and intrusion detection using packet sniffer," in *2010 Second International Conference on Communication Software and Networks*, pp. 313–317, 2010.

[4] I. Bansah, T. Montana, and S. Brako, "Implementation of intrusion detection system and traffic analysis – a case of a linux platform," *International Journal of Computer Applications*, vol. 141, pp. 21–29, 05 2016.

[5] M. Aljanabi, M. A. Ismail, and A. H. Ali, "Intrusion detection systems, issues, challenges, and needs," *International Journal of Computational Intelligence Systems*, vol. 14, pp. 560–571, 2021.

[6] N. S. Bhati and M. Khari, "Comparative analysis of classification based intrusion detection techniques," in *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 1–6, IEEE, 2021.

[7] A. A. R. Melvin, G. J. W. Kathrine, S. Pasupathi, V. Shanmuganathan, and R. Naganathan, "An ai powered system call analysis with bag of word approaches for the detection of intrusions and malware in australian defence force academy and virtual machine monitor malware attack data set," *Expert Systems*, p. e13029, 2022.

[8] A. Thakkar and R. Lohiya, "Analyzing fusion of regularization techniques in the deep learning-based intrusion detection system," *International Journal of Intelligent Systems*, vol. 36, no. 12, pp. 7340–7388, 2021.

[9] V. Kathiresan, S. Karthik, P. Divya, and D. P. Rajan, "A comparative study of diverse intrusion detection methods using machine learning techniques," in *2022 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–6, IEEE, 2022.

[10] N. Kumar and U. Kumar, "Artificial intelligence for classification and regression tree based feature selection method for network intrusion detection system in various telecommunication technologies," *Computational Intelligence*, 2022.

[11] G. Yedukondalu, G. H. Bindu, J. Pavan, G. Venkatesh, and A. SaiTeja, "Intrusion detection system framework using machine learning," in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 1224–1230, IEEE, 2021.

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

[13] R. V. Mendonça, J. C. Silva, R. L. Rosa, M. Saadi, D. Z. Rodriguez, and A. Farouk, "A lightweight intelligent intrusion detection system for industrial internet of things using deep learning algorithms," *Expert Systems*, vol. 39, no. 5, p. e12917, 2022.

[14] P. Illavarason and B. K. Sundaram, "A study of intrusion detection system using machine learning classification algorithm based on different feature selection approach," in *2019 Third international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*, pp. 295–299, IEEE, 2019.

[15] D. Katiyar, M. Tripathi, M. Kumar, M. Verma, D. Sahu, and D. Saxena, "Ai and cyber-security: Enhancing threat detection and response with machine learning," *Educational Administration Theory and Practices*, vol. 30, 2024.

[16] A. Tellache, A. Mokhtari, A. A. Korba, and Y. Ghamri-Doudane, "Multi-agent reinforcement learning-based network intrusion detection system," in *NOMS 2024 – IEEE Network Operations and Management Symposium*, pp. 1–9, IEEE, May 2024.

[17] K. Ren, Y. Zeng, Y. Zhong, and et al., "MAFSIDS: a reinforcement learning-based intrusion detection model for multi-agent feature selection networks," *Journal of Big Data*, vol. 10, no. 1, p. Article 137, 2023.

[18] S. Oesch, A. Chaulagain, B. Weber, M. Dixson, A. Sadovnik, B. Roberson, C. Watson, and P. Austria, "Towards a high fidelity training environment for autonomous cyber defense agents," in *Proceedings of the 17th Cyber Security Experimentation and Test Workshop (CSET '24)*, (New York, NY, USA), pp. 91–99, Association for Computing Machinery, 2024.

[19] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 357–370, 2022.

[20] K. C. Santos, R. S. Miani, and F. de Oliveira Silva, "Evaluating the impact of data preprocessing techniques on the performance of intrusion detection systems," *Journal of Network and Systems Management*, vol. 32, no. 2, p. 36, 2024.

[21] M. G. Lima, A. Carvalho, J. G. Álvares, C. E. d. Chagas, and R. R. Goldschmidt, "Impacts of data preprocessing and hyperparameter optimization on the performance of machine learning models applied to intrusion detection systems," *arXiv preprint arXiv:2407.11105*, 2024.

[22] D. Brzezinski, "Fibonacci and k-subsecting recursive feature elimination," *arXiv preprint arXiv:2007.14920*, 2020.

[23] M. López-De-Castro, A. García-Galindo, and R. Armañanzas, "Conformal recursive feature elimination," *arXiv preprint arXiv:2405.19429*, 2024.

[24] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between multinomial and bernoulli naïve bayes for text classification," in *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pp. 593–596, IEEE, 2019.

[25] M. Ontivero-Ortega, A. Lage-Castellanos, G. Valente, R. Goebel, and M. Valdes-Sosa, "Fast gaussian naïve bayes for searchlight classification analysis," *Neuroimage*, vol. 163, pp. 471–479, 2017.

[26] W. S. Noble, "What is a support vector machine?," *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[27] C. Zhang and Y. Ma, *Ensemble machine learning: methods and applications*. Springer, 2012.

[28] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[29] S. P. Thirimanne, L. Jayawardana, L. Yasakethu, P. Liyanaarachchi, and C. Hewage, "Deep neural network based real-time intrusion detection system," *SN Computer Science*, vol. 3, no. 2, p. 145, 2022.

[30] R. Kimanzi, P. Kimanga, D. Cherori, and P. K. Gikunda, "Deep learning algorithms used in intrusion detection systems–a review," *arXiv preprint arXiv:2402.17020*, 2024.

[31] M. Alavizadeh *et al.*, "A deep q-network (dqn) approach for network intrusion detection," *arXiv preprint arXiv:2111.13978*, 2021.

[32] A. Nair *et al.*, "Massively parallel methods for deep reinforcement learning," *arXiv preprint arXiv:1507.04296*, 2015.