Adaptive Feature Engineering via Attention-based LSTM towards High Performance Reconnaissance Attack Detection

Hamidah Alanazi*, Shengping Bi†, Tao Wang† and Tao Hou‡
*Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia, hnalanazi@pnu.edu.sa
†New Mexico State University, Las Cruces, NM, USA, {sbi, taow}@nmsu.edu
‡Texas State University, San Marcos, TX, USA, taohou@txstate.edu

***Abstract*—The evolution of the next-generation network has indeed facilitated seamless communication and coexistence of diverse devices with varying capabilities and connectivity requirements in modern communication systems. However, it also incurs a wide range of network attacks (e.g., DDoS, U2R, R2L, and Reconnaissance attacks). Among them, reconnaissance attacks seem receiving less attention as they are passively launched and do not directly impact the network. Nevertheless, reconnaissance attacks could serve as a critical preliminary step of advanced attacks and should never be left behind. In this work, we develop a novel neural learning framework that incorporates feature selection and model training together to achieve a better performance in reconnaissance attack detection. In particular, our framework contains two key techniques: 1) Self-adaptive feature selection integrates the attention and dropping mechanisms to facilitate dynamic feature set adaption. 2) Lightweight attention-based LSTM for feature learning. It allows us to attend to relevant packets within each traffic flow and extract the temporal dependency of reconnaissance behaviors for better attack identification. In the experiment, we build a real-world network testbed to validate our design. The results show that the detection model can achieve a detection rate of up to 99.88% with the proposed neural learning detector.**

***Index Terms*—Reconnaissance Attack, LSTM, Attention, Feature Engineering, Intrusion Detection**

I. Introduction

The evolution of the next-generation network has indeed facilitated seamless communication and coexistence of diverse devices with varying capabilities and connectivity requirements in modern communication systems. Nevertheless, the proliferation of diverse network-connected devices also creates subtle attack surfaces for various network attacks, such as Distributed Denial-of-Service (DDoS) attacks, User-to-Root (U2R) attacks, Remote-to-Local (R2L) attacks, and Reconnaissance attacks. While advanced attacks that can explicitly compromise the systems or disrupt the network operations (e.g., DDoS, U2R, R2L) have been intensively investigated by various studies, reconnaissance attacks seem receiving less attention as they are passively launched and do not directly impact the network [1].

Nevertheless, reconnaissance attacks could serve as a critical preliminary step of advanced attacks and should never be left behind. By exploring open ports of a communication system, monitoring online activities of network critical infrastructures, or gathering the sensitive information of a target network, reconnaissance attacks allow adversaries to identify the potential security risks and weakness point of the target systems, which further opens a door for exacerbating their malicious purpose. For example, an attacker can leverage the probe attack, which is a kind of reconnaissance attack, to identify the most critical infrastructures within the network and craft an advanced DDoS attack specifically targeting those critical infrastructures [2]. According to Cisco [3], reconnaissance attacks are expected to expand at a compound annual growth rate of 12.3% from 2021 to 2028 due to the increasing number of network-connected devices. There are growing needs to develop effective and efficient detection mechanisms to mitigate the threats posed by reconnaissance attacks.

Recently, researchers have been exploring various techniques to detect different network attacks [4]. The typical techniques include signature-based detections, protocol analyses, and machine learning-based detections [5]. Signature-based detections rely on predefined patterns or signatures (e.g., specific IP addresses, ports, or protocols commonly used in attacks) to identify malicious activities. While they are effective at identifying known attacks, they may not be effective when it comes to detect new or unknown attacks that do not match any existing patterns or signatures. Protocol analyses involve examining the content and structure of network protocols (e.g., packet headers and payloads) to identify deviations or abnormalities that may indicate an attack. However, such a technique requires detailed knowledge of the specific protocols being used, including their expected behavior and message formats, which may be challenging for unclear new attacks when protocols are not well-documented or not well-understood.

Nowadays, machine learning has emerged as a powerful and widely adopted technique for network attack detection [5, 6]. Various machine learning based detections (e.g., Naive Bayes, Random Forest, KNN, RNN, and MLP) have been developed and can achieve a good performance. By analyzing features extracted from the network traffic, machine learning can build models to characterize the traffic patterns and further identify different types of network attacks. With the capability of automatic learning to adapt new attack patterns, machine learning based detections offer improved flexibility and reliability compared to traditional detections.

Intuitively, we can simply adopt existing machine learning based network attack detectors for reconnaissance identification. However, our initial evaluation reveals significant variations in the performance of these detectors when using different feature sets from various datasets (e.g., KDDCUP99, NSL-KDD, UNSW-NB15, etc). Since reconnaissance attacks are typically launched stealthily within low-volume and low-frequency network flows, the accuracy and reliability of a machine learning based detector heavily relies on the quality of features. A high-quality feature set can essentially contribute to the efficiency of model training and improve the detection rate, while a feature set lacking representativeness may lead to training overhead and false detections. In this work, we develop a novel neural learning framework that incorporates feature selection and model training together to achieve a better performance in reconnaissance attack detection. In particular, our framework contains two key techniques:

(i) Self-adaptive feature selection: Typical feature selection algorithms, such as Chi-square test, Fisher's score, random forest importance, are proceeded as an independent process and do not cooperate with the model training process. They may not select the optimal feature set that aligns with the specific machine learning algorithms being used, and lack the feasibility for online incremental feature set revision. To address the issue, we incorporate a self-adaptive feature selection layer within the model training. In particular, the feature selection layer integrates both attention and dropout mechanisms, where attention is to identify the most critical and relevant features allowing the model to focus on the most informative aspects of the data, and dropout is to remove the redundant features to reduce the computational overhead and improve the training efficiency. In addition, by incorporating feature selection within the model training process, we enable the model to dynamically tune the feature set to the specific machine learning algorithms employed and accommodate network dynamics.

(ii) Lightweight attention-based LSTM for feature learning: Conventional feature extraction typically takes place at the data preprocessing, which may not allow the machine learning model to learn new features to accommodate network dynamics and emerging attacks during training stage. To address the limitation, we propose a novel attention driven LSTM network that can learn feature representations and model the temporal dependencies during the training stage, specifically for reconnaissance attacks that involve a series of probe packets with varying lengths and timings. In particular, we tailor-make a lightweight LSTM network that can effectively characterize the packet correlations of incoming traffics, yet substantially reduce the training overhead. In addition, as incoming traffic consists of both attack and benign packets, it is important to distinguish them as different traffic flows. To achieve this, we employ an additional attention layer that allows the model to selectively attend to relevant packets within each traffic flow, thus effectively learning the flow-based features necessary for accurate attack identification.

We also build a real-world network testbed to validate our design. The results show that the detection model can achieve a detection rate of up to 99.88% with the proposed neural learning detector.

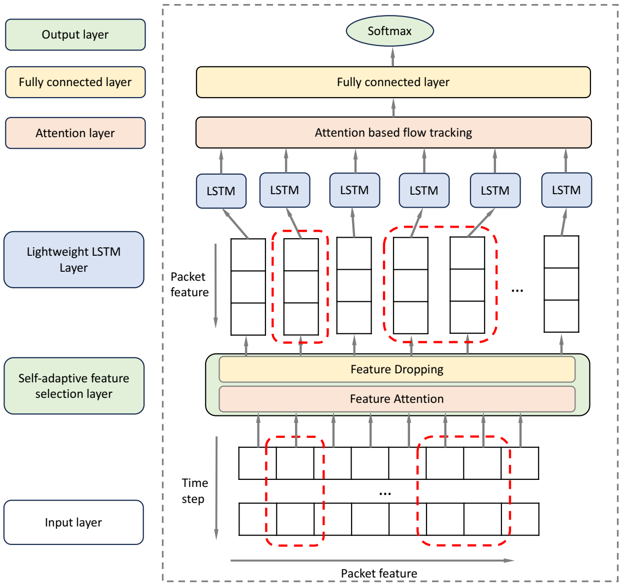

Fig. 1. Attention-based LSTM for Feature Engineering.

II. Proposed Method

Feature learning and model training are two key stages for reconnaissance attack identification. Traditional techniques usually split them as two independent steps, allowing domain experts to focus on their respective areas of specialization. Nevertheless, these methods heavily rely on manual feature engineering, which can be time-consuming, domain-dependent, and may not capture all the relevant information in the data. In this section, we propose a novel neural learning framework that incorporates both feature learning and model training to gain a better understanding of reconnaissance behaviors and further improve the detection rate.

A. Model Overview

As shown in Figure 1, the proposed neural learning framework incorporates an integrated approach for feature selection and model training. Instead of treating feature selection as a separate pre-processing step, the framework leverages the power of neural networks to dynamically adapt the feature set and learn relevant features directly in the training process. Two key techniques have been proposed: Self-adaptive feature selection incorporation and Attention driven Light LSTM feature learning.

- **Self-adaptive feature selection:** The technique integrates the attention and dropout mechanisms to facilitate dynamic feature set adaption. The attention mechanism is capable of assessing feature importance, allowing the model to focus on the most informative aspects of the data. The dropout mechanism is to remove the redundant features to reduce the computational overhead and improve the training efficiency.
- **Lightweight attention-based LSTM for feature learning:** It contains two components: 1) A tailored LSTM

network is developed to understand the temporal dependency of reconnaissance behaviors for better attack identification. 2) An attention based flow tracking is proposed allowing us to attend to relevant packets within each traffic flow, and effectively extract the flow-based features.

The objective of the proposed neural learning framework is to better understand the traffic patterns of reconnaissance attacks, yet substantially reduce the training overhead. We describe each technique in the following sections.

B. Self-adaptive Feature Selection

Feature selection aims to choose the most relevant and informative features from a feature pool, allowing for more effective and efficient model building. Traditional feature selection algorithms such as Chi-square test, Fisher's score, random forest importance, are proceeded as an independent step from the model training. They may not be able to identify the optimal feature set that aligns with the specific machine learning algorithm and lack the feasibility for online feature set revision.

Instead of adopting independent feature selection algorithms, we propose a self-adaptive feature selection scheme that incorporate feature selection within the model training for improved flexibility and efficiency. Two different layers (i.e., attention and dropout) have been deployed in order to achieve efficient dynamic feature selection. Figure 2 depicts the detailed structure of the proposed feature selection technique.

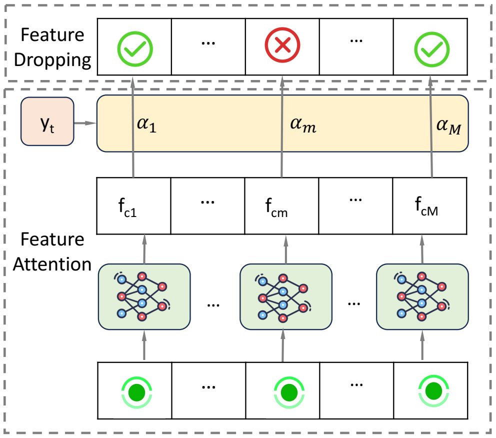

Fig. 2. Self-adaptive Feature Selection.

1) Attention layer: Attention mechanism allows us to focus on the most informative features relevant to the attack detection. By assigning attention weights to different features, the model can effectively learn to distinguish between benign and reconnaissance traffics.

As shown in Figure 2, the attention mechanism assigns a shallow neural network to each input feature to assess their correlation with the reconnaissance attacks. Particularly, each shallow network will take instances of a particular feature as input and generates an output vector f_i indicating the correlation between the feature and the target. The attention weight is then calculated as a function of the classification result y_t and output vector f_i.

$$\alpha_i = \frac{exp(Corr(y_t, f_i))}{\sum_{i=1}^{M} exp(Corr(y_t, f_i))}, \tag{1}$$

where $Corr(y_t, f_i)$ is the weighted dot product between y_t and f_i, denoted as $y_t^T W_\alpha f_i$.

The attention layer, in conjunction with the Light LSTM detector, is updated using the back propagation mechanism. Particularly, by propagating the gradients backward through the model, the attention weights can be updated iteratively. It enables the model to continuously update attention weights, establishing the most accurate description of the relevance between the features and attacks.

2) Feature dropping layer: Feature dropping is to remove redundant features and keep the most effective feature set according to the attention weights from previous layer. We employ a dropping threshold to adaptively balance the detection rate and the computational overhead. In our scheme, the dropping threshold is empirical determined to maintain a feature set with around 20 features.

C. Lightweight Attention-based LSTM for Feature Learning

As reconnaissance attacks usually involve a series of probe packets with varying lengths and timings, it's crucial to understand the temporal dependency and flow patterns of the reconnaissance traffic for effectively identifying and analyzing such attacks. Specifically, when a packet arrives, it is essential to go beyond extracting only packet-based features. More importantly, identifying the traffic flow to which each packet belongs and extracting the corresponding flow-based features are critical steps in analyzing reconnaissance attacks. To this end, we propose the lightweight attention-based LSTM for feature learning. The customized LSTM is to analyze the sequence of packets and understand the temporal patterns and dependencies inherent in the reconnaissance traffic, while the attention based flow tracking allow us to attend on packets belong to the same flow only.

1) Lightweight LSTM: We conduct a real-time attack detection, indicating that we don't wait for the entire traffic flow to be received before analyzing it. Alternatively, when a new packet arrives, we will examine it with previous received packets and determine the traffic flow to which the new received packet belongs. This allows us to extract the flow features as soon as a new packet has been received and enable real-time attack identification. To control the scope of examination, a time window is employed to determine the maximum number of previous packets to be examined.

We employ LSTM to capture temporal dependencies and extract flow-based features due to its capability to handle long-term dependencies and process sequential data. The number of memory cells is determined by the packet window. In our scheme, we adopt 20 memory cells to examine 20 previous received packets in total. In addition, we have tailored the LSTM for efficient flow based feature extraction. Figure 3 illustrates the customized LSTM used in our scheme.

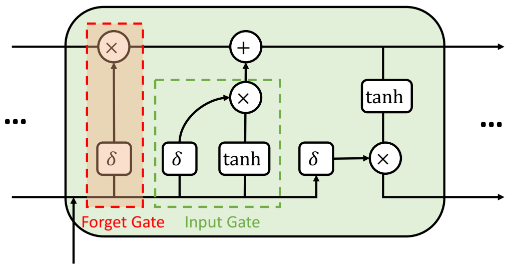

Fig. 3. Lightweight LSTM

- Forget gate: It is to determine whether to retain or discard information from the previous time step. In flow-based feature extraction, all packets belonging to the same flow are considered equivalent, and it is important to accumulate the flow features to achieve accurate temporal packet correlation. To accomplish this, we freeze the forget gate by setting its value as 1, indicating that the features extracted from each packet will be fully retained without any information loss.
- Input gate: It is to assess and quantify the significance of the new information conveyed by the input. In flow feature extraction, the information of incoming packet will be retained if it belongs to the examined traffic flow. Otherwise, the input will be discarded. To achieve this, we modify the input function of weight parameters and bias using a weighted correlation formula. This formula helps determine the probability that a packet belongs to the examined flow, allowing for selective retention or discarding of input features. In particular, the input gate is formulated as follows:

$$\begin{aligned} i_t &= \delta(W_i.[h_{t-1}, x_t] + b_i) \\ \hat{C}_t &= tanh(W_c.[h_{t-1}, x_t]) + b_C, \end{aligned}$$

where $W_i.[h_{t-1}, x_t]$ is the weighted correlation, x_t is the input at time t, h_{t-1} is the hidden state at time $t-1$, and W and b are the weight parameters and biases.

We make specific modifications to the LSTM model to enhance flow feature extraction while reducing training overhead. First, we freeze the forget gate, ensuring that the information from the previous time step is fully retained and reduce the training overhead. In addition, we customize the input gate function to improve flow feature extraction.

2) Attention based flow tracking: To facilitate the accurate extraction of flow-based features and differentiate between various traffic flows, we incorporate an additional attention layer on top of the LSTM. This attention layer enables us to identify and focus on packets belonging to the same traffic flow, thus enhancing the precision of feature extraction.

The computation of attention weights for each input packet is similar as section II-B1. By iteratively updating these attention weights using back propagation, we aim to train the model to effectively identify the traffic flow and assign appropriate attention to each packet.

III. Data Set Description

In this section, we introduce the datasets used in this work. We also build a real-world network testbed and collect our own custimized dataset to further validate our design.

A. KDDCUP99 Dataset

Tthe dataset KDDCUP99 [7] is used for our feature selection and initial model training. The dataset includes a wide variety of network attacks (e.g., DoS, R2L, U2R, Probing) simulated in a military network environment. In our study, we are particularly interested in probing attacks, which aim to extract privacy information and find potential vulnerabilities via system and network scanning. We take a random sampling (i.e., 41,102 overall samples) of probing/normal traffics from the dataset, each of which comes with 41 features, including basic features for individual TCP connections, traffic features within a certain time window, and content features relevant to specific domain knowledge.

B. NSL-KDD Dataset

The NSL-KDD [7] dataset is a benchmark dataset widely used for the design and evaluation of intrusion detection systems (IDS). It is an improvement over the original KDD Cup 1999 dataset. The NSL-KDD dataset was developed by selecting a subset of the original KDD Cup 1999 dataset and addressing its shortcomings. The redundant and irrelevant features were removed, and additional network traffic instances were added to increase the diversity and realism of the dataset. The NSL-KDD dataset consists of approximately 5 million network connection records with 41 features. These records are classified into two categories: normal and attack. The attack category is further divided into three subcategories: denial-of-service (DoS), user-to-root (U2R), and remote-to-local (R2L).

C. UNSW-NB15 Dataset

The UNSW-NB15 [8] dataset is a benchmark dataset for network security design and evaluation. It was created by the University of New South Wales (UNSW) in Australia and consisted of network traffic data collected in a controlled environment. The dataset contains network traffic captures representing nine different types of attacks and normal network traffic. The attacks include various types of intrusion attacks, such as Denial-of-Service (DoS), Distributed DoS (DDoS), and probing attacks. The normal traffic is collected from regular network activities. It contains approximately 2.5 million instances, and 49 features represent each instance in the dataset.

D. Customized Dataset from Our Real-world Testbed

In order to further validate our model, we build a local network testbed for real-world data collection. Our network consists of workstations, laptops, mobile devices, and IoT devices. All devices are connected to a 2.4 GHz WiFi router to form a client-and-server LAN. In particular, we setup an Apache HTTP Server in one workstation, which hosts a personal site as the target of probing attacks. All other devices are served as the clients, and the attacker is one of them. The real-time data traffic is collected and saved as a PCAP file. our dataset contains 69,000 samples, of which 34,500 are probing attacks and the others are benign traffics. The real-world dataset will help us to further evaluate the proposed model's effectiveness.

IV. Experimental Evaluation

In this section, we evaluate the effectiveness of the proposed design using multiple datasets and compare its performance with different machine-learning algorithms.

A. Effectiveness of Cross-layer Attention Design

In our work, we have adopt two attention layers for feature selection and traffic flow tracking. In feature selection, attention mechanism is to identify most important feature for attack detection. Figure 4 plots the heat map of attention weights for different features across four datasets. As shown, with different datasets, their feature attention weights will change accordingly. The observation indicates that self-adaptive feature selection is indeed needed to accommodate network dynamics for better attack identification.

In traffic flow tracking, attention layer is inserted to identify traffic flows to which each packet belongs to. In figure 5, we plot attention weights of packets from 20 different packet sequences. As shown, each sequence is mixed with different traffic flows. Attention mechanism is indeed required to distinguish different flows and extract exact flow based features.

B. Effectiveness of the Proposed Detector

To validate our design and demonstrate its effectiveness. We compare the performance of our detector with following four machine learning techniques.

- **Random Forest** is an ensemble learning classifier based on decision trees. It collects votes from various decision trees and determine the best final class by taking the average from these decision trees [9].
- **Naïve Bayes** Naïve Bayes is a supervised machine learning algorithm that makes predictions by leveraging the conditional independence assumption among features [10].
- **XGBoost** is a machine learning algorithm that utilizes parallel tree boosting techniques to improve model performance [11].

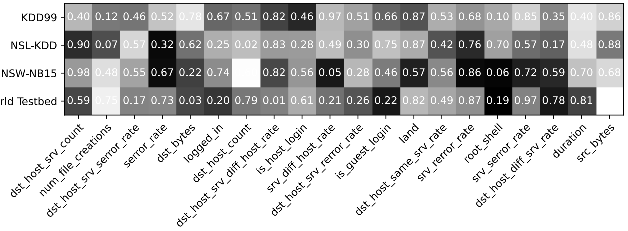

Fig. 4. Heat Map for Feature Attention.

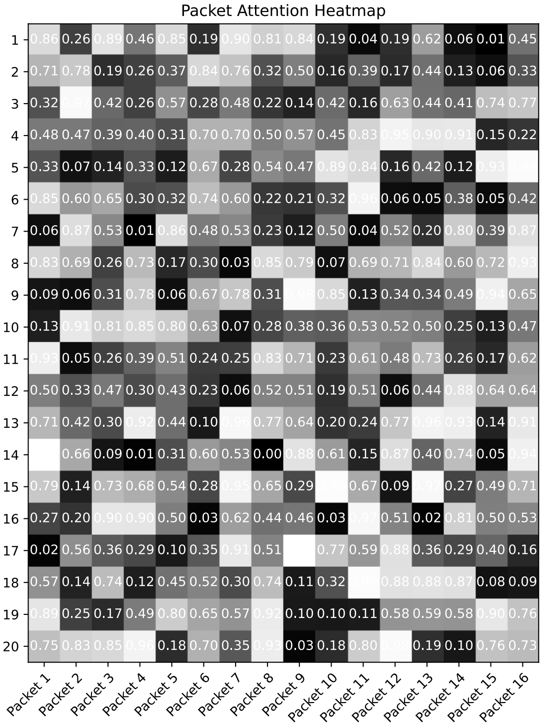

Fig. 5. Heat Map for Packet Attention.

- **K-Nearest Neighbors (KNN)** is a supervised learning algorithm used for classification and regression tasks. In KNN, the predicted class or value for a new data point is determined by considering the majority class or average value among its k nearest neighbors [12].

The above machine learning algorithms show good performance in traffic identification. We will validate our design by comparing its performance with all of them.

C. Evaluation Metrics

We use following metrics to evaluate the effectiveness of the proposed model.

- **Detection Accuracy:** It is defined as the ratio of the number of accurate detections to all the detections.
- **Precision:** It is defined as the ratio of true positives (TP) to the sum of true positives and false positives.
- **Recall:** It is defined as the proportion of true positive predictions among all the actual positive examples in the dataset.
- **F1 Score:** it is the harmonic mean of precision and recall, providing a balanced measure that considers both the ability of the model to correctly identify positive instances and the ability to capture all positive instances, i.e., $F1 = 2 * (Precision * Recall)/(Precision + Recall)$.

D. Performance Evaluation on Different Datasets

We evaluate and compare the performance of our proposed detection scheme with five commonly used machine learning algorithms on different datasets. The results, presented in

Table I, Table II, Table III, and Table IV, indicate that our lightweight attention-based LSTM consistently achieves a detection rate of over 96.52% across all four datasets.

Furthermore, the experimental evaluation demonstrates that our proposed lightweight attention-based LSTM outperforms the five machine learning algorithms considered in our study. For instance, in Table IV, our scheme achieves a detection rate of 99.10%, while LSTM achieves 94.83%, Random Forest achieves 94.32%, Naive Bayes achieves 81.42%, XGBoost achieves 94.28%, and KNN achieves 94.70%. We can observe a trivial improvement of the proposed neural learning technique.

TABLE I
EVALUATION RESULTS FOR KDDCUP99 DATASET

ML Name	Accuracy	precision	recall	f1 score
LSTM	95.92 %	0.94	0.95	0.94
Attention Driven LSTM	**99.88%**	**0.99**	**0.99**	**0.99**
Random Forest	95.32%	0.94	0.95	0.94
Naive Bayes	96.06 %	0.96	0.96	0.94
XGBoost	96.02%	0.92	0.96	0.94
KNN	93.11%	0.95	0.93	0.94

TABLE II
EVALUATION RESULTS FOR NLS KDD DATASET

ML Name	Accuracy	precision	recall	f1 score
LSTM	95.05 %	0.94	0.95	0.94
Attention Driven LSTM	**98.61%**	**0.98**	**0.99**	**0.98**
Random Forest	83,58%	0.86	0.84	0.84
Naive Bayes	76.97 %	0.80	0.77	0.76
XGBoost	86.12%	0.46	0.68	0.55
KNN	71.85%	0.80	0.72	0.71

TABLE III
EVALUATION RESULTS FOR UNSW-NB 15 DATASET

ML Name	Accuracy	precision	recall	f1 score
LSTM	92.44%	0.95	0.95	0.95
Attention Driven LSTM	**96.52%**	**0.95**	**0.98**	**0.96**
Random Forest	88.25%	0.88	0.90	0.88
Naive Bayes	77.71%	0.77	0.78	0.68
XGBoost	79.25%	0.80	0.79	0.79
KNN	72.20%	0.80	0.72	0.71

TABLE IV
EVALUATION RESULTS FOR TESTBED DATASET

ML Name	Accuracy	precision	recall	f1 score
LSTM	94.83%	0.95	0.95	0.95
Attention Driven LSTM	**99.10%**	**0.99**	**0.99**	**0.99**
Random Forest	94.32%	0.92	0.94	0.92
Naive Bayes	81.42%	0.95	0.81	0.86
XGBoost	94.28%	0.91	0.94	0.92
KNN	94.70%	0.96	0.95	0.95

V. CONCLUSION

In this work, we develop a novel neural learning framework that incorporates feature selection and model training together to achieve a better performance in reconnaissance attack detection. In particular, our framework contains two key techniques: 1) Self-adaptive feature selection. 2) Lightweight attention-based LSTM for feature learning. The experimental evaluation demonstrate that the proposed scheme can effectively detect the reconnaissance attack and achieve a detection rate of up to 99.88%.

VI. ACKNOWLEDGEMENT

The work was sponsored by the DEVCOM Analysis Center under Cooperative Agreement Number W911NF-22-2-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] M Uma and Ganapathi Padmavathi. A survey on various cyber attacks and their classification. *International Journal of Network Security*, 2013.

[2] Tao Hou, Zhe Qu, Tao Wang, Zhuo Lu, and Yao Liu. Proto: Proactive topology obfuscation against adversarial network topology inference. In *IEEE Conference on Computer Communications*, 2020.

[3] Grand View Research. Network probe market share analysis report, 2021-2028. *Grand View Research*, 2021.

[4] Iftikhar Ahmad, Azween B Abdullah, and Abdullah S Alghamdi. Application of artificial neural network in detection of probing attacks. In *IEEE Symposium on Industrial Electronics & Applications*, 2009.

[5] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. *Expert systems with applications*, 2009.

[6] Hamidah Alanazi, Shengping Bi, Tao Wang, and Tao Hou. Exquisite feature selection for machine learning powered probing attack detection. *IEEE International Conference on Communications*, 2023.

[7] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *IEEE symposium on computational intelligence for security and defense applications*, 2009.

[8] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems. In *IEEE military communications and information systems conference*, 2015.

[9] Reneilson Santos, Danilo Souza, Walter Santo, Admilson Ribeiro, and Edward Moreno. Machine learning algorithms to detect ddos attacks in sdn. *Concurrency and Computation: Practice and Experience*, 2020.

[10] Shenglei Chen, Geoffrey I Webb, Linyuan Liu, and Xin Ma. A novel selective naïve bayes algorithm. *Knowledge-Based Systems*, 2020.

[11] Bingyue Pan. Application of xgboost algorithm in hourly pm2. 5 concentration prediction. In *IOP conference series: earth and environmental science*, 2018.

[12] Kashvi Taunk, Sanjukta De, Srishti Verma, and Aleena Swetapadma. A brief review of nearest neighbor algorithm for learning and classification. In *IEEE International Conference on Intelligent Computing and Control Systems*, 2019.