Evaluation of Machine Learning Models for Feint Shrew Attack Detection with Simulated Network Traffic

Shingo Imai*, Tomotaka Kimura*, and Jun Cheng*
*Graduate School of Science and Engineering, Doshisha University, Kyoto, Japan
Email: ctwk0120@mail4.doshisha.ac.jp; tomkimur@mail.doshisha.ac.jp; jcheng@ieee.org

***Abstract*—This study investigated the effectiveness of machine learning techniques for detecting feint shrew attacks. A shrew attack exploits the retransmission timeout mechanism of the TCP protocol by transmitting intermittent bursts of high-rate packets, which severely degrade communication performance despite its minimal use of traffic, and is regarded as a representative type of low-rate denial-of-service (LDoS) attack [1]. In recent years, the feint shrew attack, a derivative form of the shrew attack, has emerged, making detection even more challenging by distributing traffic across multiple terminals or inserting feint traffic that closely resembles normal communication. To analyze such attacks, we constructed a network traffic dataset using ns-3 simulations and applied two machine learning models, a feedforward neural network (NN) and long short-term memory (LSTM), to classify normal and malicious traffic. The detection performance of these models was compared to evaluate the effect of considering temporal dependencies on feint shrew attack identification. The experimental results show that the LSTM model achieves higher detection accuracy than the NN model, demonstrating that incorporating temporal features is effective in detecting feint shrew attacks and highlighting the potential of machine learning-based analysis for understanding and mitigating LDoS attack behavior.**

***Index Terms*—Low-rate distributed denial-of-service, Machine learning, Deep learning, ns-3**

I. Introduction

With the rapid expansion of Internet-based social and economic activities, the availability and reliability of online services have become increasingly critical. The widespread adoption of electronic commerce, cloud computing, and the Internet of Things (IoT) has made these services essential to both business operations and social infrastructure. Consequently, service disruptions can have a severe impact on enterprises and society as a whole. Among the various cyber threats, denial-of-service (DoS) attacks, in which excessive requests or massive volumes of data are sent to disrupt normal service operation, remain one of the most serious concerns. In particular, the sophistication of attack techniques and the exploitation of botnets have made DoS attacks more difficult to detect and have led to damage on an increasingly large scale.

Recently, low-rate DoS (LDoS) attacks have emerged as a more insidious variant of DoS attacks. Unlike conventional high-rate attacks that rely on overwhelming traffic volumes, LDoS attacks exploit protocol-level vulnerabilities to intermittently transmit short bursts of malicious packets. Although their average traffic rate is low, these attacks can effectively degrade network performance by synchronizing with protocol mechanisms such as TCP congestion control or retransmission timeouts (RTOs). Because of their low traffic intensity and bursty nature, LDoS attacks are difficult to detect using conventional traffic volume-based or threshold-based methods.

Among the various LDoS variants, the shrew attack is a representative and particularly dangerous form because of its stealthiness. By periodically sending short bursts of packets that exploit TCP vulnerabilities, the shrew attack can severely degrade communication performance while remaining undetected for extended periods [2]. Consequently, it poses a significant threat to the stability of networked systems and calls for urgent countermeasures.

To address this problem, several studies have proposed detection methods based on machine learning [3] [4] and statistical analysis [5]. These approaches extract characteristic patterns from network traffic and employ anomaly detection or classification models to determine the presence of attacks. Although existing methods have achieved high detection accuracy and demonstrated effectiveness for the early detection of shrew attacks, more sophisticated shrew-attack variants have recently emerged. These evasive shrew attacks further obscure detection by modifying temporal and statistical features, presenting new security challenges.

For example, coordinated attacks involving multiple compromised devices have been proposed [6] in which the average traffic per device is significantly reduced, making their detection even more difficult than the detection of conventional LDoS attacks. Moreover, feint shrew attacks have been reported [7] in which benign-looking traffic is intentionally mixed with malicious bursts to deceive conventional detectors. Such attacks are particularly challenging to detect using only basic statistical features or aggregated traffic patterns. By contrast, machine learning-based anomaly detection models have the potential to capture the complex temporal and statistical characteristics of network traffic, making them suitable for detecting sophisticated attacks such as feint shrew attacks. However, collecting real network traffic that contains LDoS or feint shrew attacks is challenging. These attacks occur intermittently and at low rates, making packet capture and accurate labeling extremely difficult, and thus it is challenging to obtain sufficient training data for machine learning. Consequently,

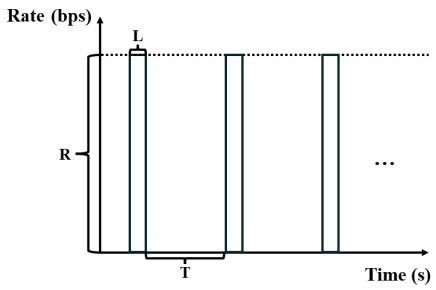


Fig. 1: Typical Shrew Attack

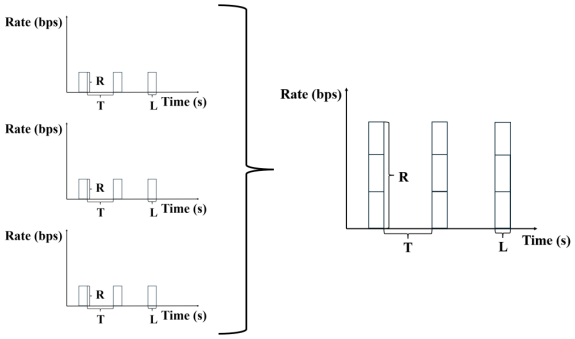


Fig. 2: Collaborative Shrew Attack

most existing studies rely on synthetic or idealized traffic, and evaluations using realistic network conditions remain limited.

In this study, we reproduce a feint shrew attack on the ns-3 simulator and construct a dataset that consists of realistic attack traffic. Using the generated time-series data, we evaluate whether machine learning models can effectively classify and detect LDoS attacks. Specifically, using the collected dataset, we apply both a simple neural network (NN) and a long short-term memory (LSTM) model, which is effective for time-series data, to clarify whether machine learning can successfully distinguish feint LDoS attacks.

The remainder of this paper is organized as follows. Section II introduces the shrew attack and its derivative, the feint shrew attack. Section III describes the methodology and experimental procedure. Section IV presents the experimental results and performance analysis. Finally, Section V concludes the paper and discusses future work.

II. Shrew Attacks and Feint Shrew Attacks

A. Shrew Attacks

Figure 1 illustrates a typical realization of a shrew attack. The attack exploits the behavior of TCP's RTO mechanism to dramatically reduce effective transmission rates. As depicted, the attacker emits short, high-rate bursts of traffic (at rate R) lasting L seconds at regular intervals of T seconds. Each burst induces temporary congestion in the network and provokes packet losses within the affected TCP flows. To maximize impact, a classical shrew attack aims to synchronize the burst interval T with the TCP RTO timing. Because TCP exponentially backs off the RTO after each timeout, an idealized shrew design would double the burst interval T on each subsequent attack so that bursts repeatedly coincide with progressively larger RTOs. Under such conditions, retransmissions are repeatedly deferred and lost, leading to connection stalls or severe throughput degradation.

However, TCP implementations such as CUBIC and NewReno employ fast retransmit and fast recovery mechanisms that undermine the practical effectiveness of an RTO-synchronized attack. When a single packet is lost and the receiver issues three duplicate ACKs, the sender retransmits the missing packet immediately, before the RTO expires, and reduces the congestion window by roughly one half; as a result, the connection can continue without waiting for an RTO. Consequently, inducing an RTO in the first place now requires multiple consecutive packet losses to be forced in the initial attack burst, a condition that is difficult to satisfy in practice.

Because of these protocol-level defenses, this study does not consider the classical RTO-synchronized shrew attack but instead focuses on a variant in which the burst interval T is fixed. In this fixed-interval shrew attack, the primary objective is not to trigger RTO-based packet drops but rather to repeatedly provoke congestion events that cause a fast retransmit and the associated halving of the sender's transmission rate. By repeatedly inducing cwnd reductions, the attacker can steadily degrade communication quality even against contemporary TCP stacks that implement fast retransmit/fast recovery.

B. Feint Shrew Attacks

A sophisticated technique to evade detection in shrew attacks is to distribute the attack across multiple collaborating hosts so that the transmission rate of each individual node is reduced, as illustrated in Figure 2. By decentralizing the attack traffic, this collaborative shrew attack reduces the likelihood of anomaly detection methods that focus on high-rate behavior from a single host. Building on this concept, a derivative known as the feint-based shrew attack has recently been introduced [7]. In this method, the attacker injects feint traffic (benign-looking packets that resemble normal communication) during the intervals between the bursts of a conventional shrew attack (that is, during the period corresponding to the burst interval T). This insertion effectively conceals the regularity of attack bursts and increases the attack's stealthiness against detection algorithms.

In the feint shrew attack, a random variable r is generated from a uniform distribution. When r exceeds a predefined threshold, feint traffic is transmitted at a rate of rR, where R denotes the fixed-interval attack rate. As a result, even during the inter-burst periods, the traffic does not become completely idle but instead exhibits fluctuations that closely resemble ordinary network activity. This randomized behavior obscures abnormal periodicity and makes it more difficult for conventional detectors relying on statistical or temporal traffic patterns to identify the attack.

III. Methodology

A. Overview

In this study, we aim to verify whether machine learning models can classify feint LDoS attacks using traffic data generated through network simulation. Our objective is to evaluate the effectiveness of existing models when applied to realistically reproduced attack traffic. The methodology consists of two main stages: (1) the reproduction of feint shrew attacks in the ns-3 simulator to generate traffic datasets, and (2) the application of machine learning models to the collected time-series data to assess their classification performance. In this section, Section III-B describes the procedure for generating the dataset using ns-3 simulations and Section III-C explains the machine learning models employed to distinguish between normal and abnormal traffic.

B. Data Generation Using ns-3 Simulation

To obtain traffic data that include feint LDoS attacks, we constructed a virtual network environment in ns-3 simulator. The network topology consists of multiple TCP and UDP users, attack nodes, relay nodes, and a central server. Attack nodes perform feint shrew attacks, transmitting intermittent high-rate bursts and inserting low-rate feint traffic during idle intervals to imitate normal communication. The simulation was executed 1000 times, and the transmitted data volume at the server was recorded every 0.05 s as time-series data. This approach enables the creation of a dataset containing both normal and attack traffic, which would be difficult to capture in real network environments.

The network structure is shown in Figure 3. The network consists of three types of senders: TCP users, UDP users, and attackers. All nodes communicate with the server via a relay node and a secondary relay node. As summarized in Table III, there are ten TCP user nodes, ten UDP user nodes, and five attacker nodes. The attackers use the UDP protocol to communicate in order to imitate the behavior of UDP users.

Each link in the network has a different bandwidth between the sender and the relay node: 5 Mbps between TCP users and the relay, 10 Mbps between UDP users and the relay, and 10 Mbps between attackers and the relay. The links between the relay and secondary relay have a transmission rate of 25 Mbps, and those between the secondary relay and the server have a transmission rate of 50 Mbps. All links have a latency of 2 ms. The detailed link configurations are summarized in Table IV.

Each communicator transmits data to the server through its own dedicated application. The TCP user communicates at a constant rate of 500 Kbps throughout the simulation. To increase the diversity of attack patterns, the parameters of the UDP user and the attacker are randomly selected for each simulation run. An overview of the traffic behavior between the UDP user and the attacker is shown in Figures 4 and 5. The corresponding parameter settings and their variation ranges are summarized in Tables I and II. In each time interval of length L, the UDP user generates a uniform random number

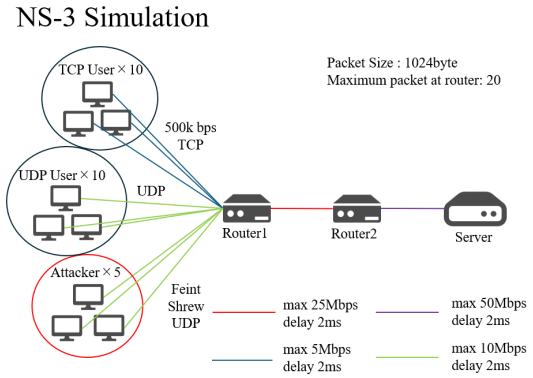


Fig. 3: Simulation Network Topology

TABLE I: Parameter Settings for UDP Users

Parameter	Range
S (s)	0-3
L (s)	0.05-0.15
R (Mbps)	2-3

$r \in [0, 1]$. If $r > 0.5$, the user transmits packets at a rate of rR; otherwise, the transmission rate is set to zero. The parameter R denotes the reference transmission rate that also serves as the basis for the attacker's traffic generation. In other words, the attacker uses the same rate parameter R_{user} as that of the legitimate UDP user within each simulation, thereby imitating normal communication behavior to conceal malicious activity. Furthermore, the attacker's burst interval L' is independently adjusted to flexibly control the timing of the attack while maintaining similarity to the UDP user's communication pattern.

Figure 6 shows the traffic traces generated by our ns-3 simulations that reproduce both a fixed-interval shrew attack and a cooperative feint shrew attack. Figure 6a depicts the temporal evolution of the transmission rate when a shrew attack is performed by a single attacker. During each burst period of length L, packets are transmitted at a fixed rate R, producing distinct high-rate spikes that are clearly observable in the trace. In contrast, Figure 6b shows the average traffic rate when the same attack behavior is distributed across 100 cooperative nodes. The red-shaded region between 0 s and 10 s indicates the interval during which individual attackers transmit their bursts; the aggregate traffic peak is slightly delayed relative to this interval because of transmission delays and queuing effects in the network. These two examples illustrate the difference between a single-node shrew attack and a distributed, feint variant, and they serve as the primary data sources for the subsequent time-series analysis and machine learning experiments described in this study.

To avoid time-dependent bias caused by synchronized attack timing across simulations, random starting intervals were used when extracting the data segments for training. Each extracted sequence was labeled as either "normal" or "attack" depending on whether it contained active attack intervals. As a result, we constructed a balanced dataset suitable for binary classification

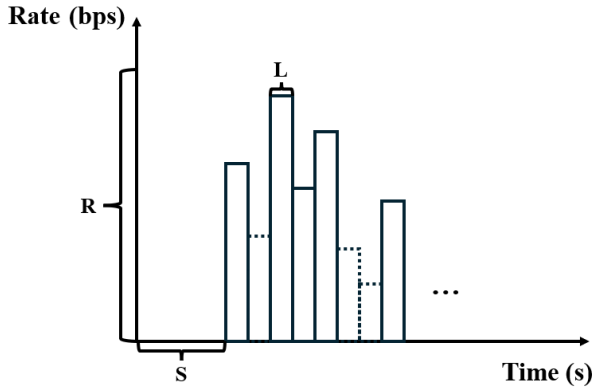


Fig. 4: UDP User Traffic

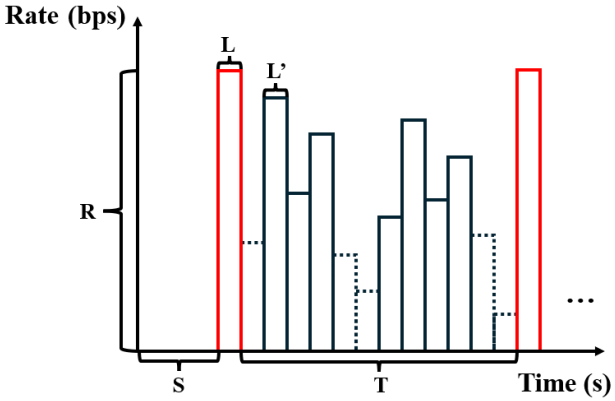
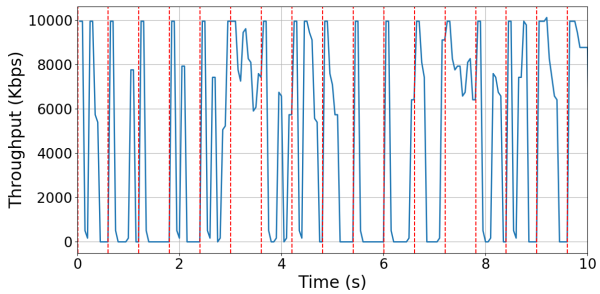


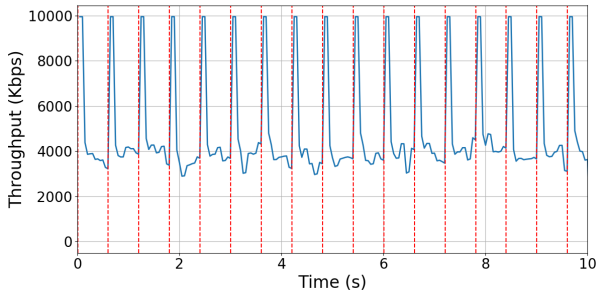
Fig. 5: Attacker Traffic

TABLE II: Parameter Settings for Attackers

Parameter	Range
S (s)	0-3
T (s)	0.2-0.5
L (s)	0.05-0.15
L' (s)	$T[\frac{L_{user}}{T}]$
R (Mbps)	R_{user}



(a) Traffic of one device



(b) Average Traffic of 100 devices

Fig. 6: Comparison of Shrew Attack Traffic Behavior Based on Number of Devices

tasks.

C. Machine Learning Models

For the classification task, two machine learning models were employed: a simple NN and an LSTM network. These two models were used to investigate whether incorporating temporal information through sequential learning can improve the ability to distinguish feint LDoS attacks from normal communication. In particular, the NN serves as a baseline model that processes static input vectors, whereas the LSTM should capture time-dependent variations that could reveal subtle attack patterns. Attempts to utilize such time-series features for detection are being widely pursued. [8] [9]

NN The NN model consists of three fully connected (Linear) layers, each followed by a ReLU activation function. A dropout layer (dropout rate = 0.2) was inserted after the first and second fully connected layers to prevent overfitting. This network captures static statistical relationships within each input vector but does not model temporal dependencies across time steps.

LSTM The LSTM model is composed of two stacked LSTM layers followed by a fully connected output layer. Each LSTM layer has 64 hidden units with a tanh activation in the cell state and sigmoid activations for the input, forget, and output gates. This architecture allows the model to learn long-term temporal dependencies within sequential data. By controlling information flow through the gating mechanism, the LSTM retains meaningful temporal features while filtering out irrelevant fluctuations, making it particularly effective for analyzing dynamic traffic patterns associated with feint LDoS attacks.

Both models take as input a time-series vector of observed traffic values. Specifically, the observed traffic volume at time t is denoted as $x(t)$, and a continuous sequence of N traffic values from t to $t + (N-1)\delta$ is represented as the vector

$$\boldsymbol{x}(t) = (x(t), x(t+\delta), \ldots, x(t+(N-1)\delta)).$$

By generating time-series traffic data over multiple time steps with and without shrew attacks, we constructed a dataset

TABLE III: Configuration of the Senders

	Number of Nodes	Protocol
TCP Users	10	TCP
UDP Users	10	UDP
Attackers	5	UDP

TABLE IV: Link Configuration

Link	Transmission Rate (Mbps)	Delay (ms)
TCP User – Router 1	5	2
UDP User – Router 1	10	2
Attacker – Router 1	10	2
Router 1 – Router 2	25	2
Router 2 – Server	50	2

$\mathcal{X}$ composed of vectors $\boldsymbol{x}(t)$. Each sample in $\mathcal{X}$ was labeled according to whether the corresponding time window represents an attack or normal communication. Using this dataset, a binary classification model f was trained to output the probability of each class (attack/normal) through two output nodes.

As a result of the specifications of the ns-3 simulation program, all shrew attacks were executed using the same timing in every simulation run. To prevent time-dependent bias arising from this synchronization, we adopted a data extraction approach that randomly selects fixed-length intervals with varying start times. This strategy removes the influence of absolute time and allows the dataset to capture the intrinsic characteristics of shrew attacks rather than artifacts of synchronization.

When constructing the training and testing datasets, data obtained from the same terminal within a single simulation were strictly separated to prevent data leakage and overfitting. To evaluate model robustness, we varied the input vector length N under six different settings, where $N \in \{11, 16, 21, 31, 41, 61\}$. Since traffic data were recorded at 0.05-s intervals ($\delta = 0.05$), these lengths correspond to time windows of approximately 0.5, 0.75, 1.0, 1.5, 2.0, and 3.0 seconds, respectively.

The architectures of both models are summarized in Tables V and VI. Each model performs binary classification to distinguish between normal and attack traffic. During training, the number of samples corresponding to UDP users and shrew attackers was balanced to mitigate class imbalance and ensure fair learning. Both models were trained using the cross-entropy loss function and optimized with the Adam optimizer (learning rate = 0.001, batch size = 32, number of epochs = 50).

IV. Evaluation Results

This section presents the experimental evaluation of machine learning models for detecting LDoS attacks. We examine and compare the detection performance of a NN and an LSTM model trained on the traffic dataset generated by ns-3 simulator. The objective of this evaluation is to investigate whether incorporating temporal dependencies through the LSTM enhances the ability of machine learning models to distinguish feint LDoS attacks from normal traffic.

TABLE V: Details of the NN Architecture

Layer No.	Description
1	Fully Connected (Linear)
	Activation Function (ReLU)
	Dropout
2	Fully Connected (Linear)
	Activation Function (ReLU)
3	Fully Connected (Linear)

TABLE VI: Details of the LSTM Network Architecture

Layer No.	Description
1	LSTM
2	LSTM
3	Fully Connected (Linear)

First, the effectiveness of the NN and LSTM models against the fixed-interval shrew attack was evaluated. Figure 7 presents the results for recall, precision, and accuracy as functions of window size N. Both the NN and LSTM models achieved high performance results at relatively small values of N, indicating that these models are effective at detecting the fixed-interval shrew attack.

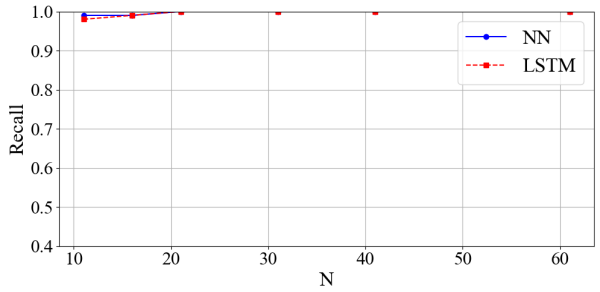
Next, the effectiveness of the machine learning models against the feint shrew attack was evaluated. Figure 8 shows the variations in recall, precision, and accuracy as functions of window size N. In contrast to its performance when detecting the fixed-interval shrew attack, the NN performed worse at small values of N, suggesting it has a limited ability to capture the temporal characteristics of the feint traffic. By contrast, the LSTM model maintained high performance even at small N, demonstrating that incorporating temporal dependencies enables more accurate discrimination between a feint shrew attack and normal communication. Furthermore, when the LSTM model was used, the accuracy was almost 1.0 for $N = 41$ and $N = 61$, indicating that the proposed approach can accurately detect feint shrew attacks from only a few seconds of input data. These results clearly confirm that employing LSTM is highly effective for detecting feint shrew attacks.

V. Conclusion

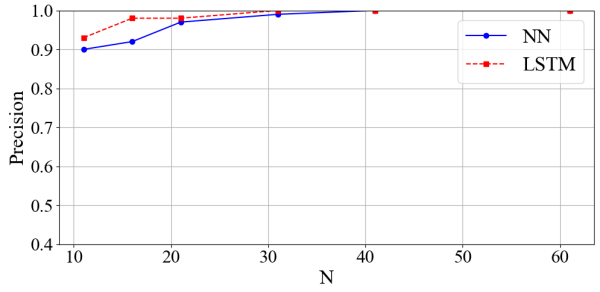
In this study, we constructed a network traffic dataset using ns-3 simulations to analyze the effectiveness of machine learning models for detecting feint shrew attacks. A shrew attack is a type of LDoS attack that can severely degrade network performance despite generating only a small amount of traffic. The feint variant, which inserts benign-like communication packets, poses an even greater challenge for conventional statistical detection methods. Using the simulated dataset, we trained and compared two models, a feedforward NN and an LSTM network, to evaluate the impact of temporal modeling on detection performance. The experimental results showed that the LSTM model achieved substantially higher detection accuracy than the NN model, demonstrating that incorporating temporal dependencies is effective for identifying feint shrew attacks.

TABLE VII: Machine Learning Training Parameters

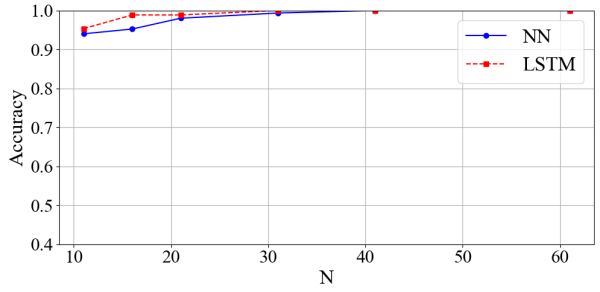
Parameter	Description
Number of Training Samples	4800
Number of Test Samples	1200
Loss Function	Cross-Entropy Error
Optimizer	Adam



(a) Recall

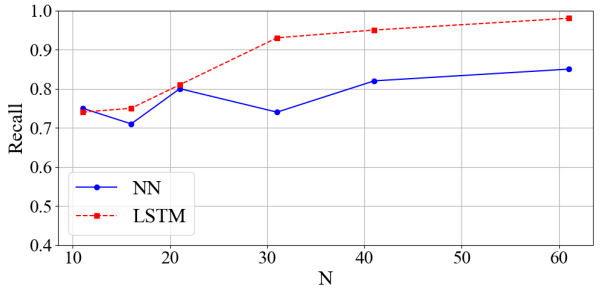


(b) Precision

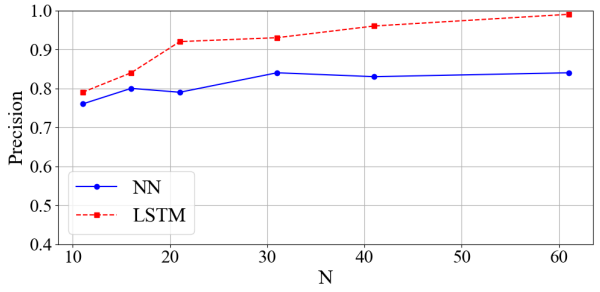


(c) Accuracy

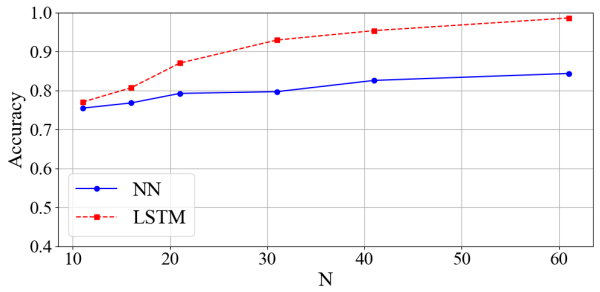
Fig. 7: Performance Metrics for the Fixed-Interval Shrew Attack.



(a) Recall



(b) Precision



(c) Accuracy

Fig. 8: Performance Metrics for the Feint Shrew Attack.

ACKNOWLEDGMENT

This research was supported by JSPS KAKENHI (23K11077).

REFERENCES

[1] A. Kuzmanovic and E. W. Knightly, "Low-rate tcp-targeted denial of service attacks: the shrew vs. the mice and elephants," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 75–86.

[2] W. Zhijun *et al.*, "Low-rate DoS attacks, detection, defense, and challenges: A survey," *IEEE Access*, vol. 8, pp. 43 920–43 943, 2020.

[3] R. Bocu and M. Iavich, "Enhanced detection of low-rate ddos attack patterns using machine learning models," *Journal of Network and Computer Applications*, vol. 227, p. 103903, 2024.

[4] A. O. M. Salih, "Exploring LDoS attack detection in sdns using machine learning techniques," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19 568–19 574, 2025.

[5] N. Gogoi *et al.*, "Shrew DDoS attack detection based on statistical analysis," *ISeCure*, vol. 16, no. 2, 2024.

[6] H. Singh *et al.*, "Shrew distributed denial-of-service attack in IoT applications: A survey," in *Internet of Things. Advances in Information and Communication Technology*, Cham, 2024, pp. 97–103.

[7] T. Cai *et al.*, "Catch me if you can: A new low-rate DDoS attack strategy disguised by feint," in *Proc. of CSCWD'23*. IEEE, 2023, pp. 1710–1715.

[8] C. Xu, J. Shen, and X. Du, "Low-rate dos attack detection method based on hybrid deep neural networks," *Journal of Information Security and Applications*, vol. 60, p. 102879, 2021.

[9] Y. Fu, X. Duan, K. Wang, and B. Li, "Low-rate denial of service attack detection method based on time-frequency characteristics," *Journal of Cloud Computing*, vol. 11, no. 1, p. 31, 2022.