Malicious Traffic classification using Long Short-Term Memory (LSTM) model

Naresh Kumar Thapa K (knt.ece@rmkec.ac.in)
RMK Engineering College
N. Duraipandian
Velammal Engineering College

Research Article

Keywords:

Posted Date: February 1st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-159180/v1

Version of Record: A version of this preprint was published at Wireless Personal Communications on March 13th, 2021. See the published version at https://doi.org/10.1007/s11277-021-08359-6.

Malicious Traffic classification using Long Short-Term Memory (LSTM) model

K. Naresh Kumar Thapa[1], N.Duraipandian[2]
[1]Research Scholar, Anna University & Assistant Professor, Dept.of.ECE, RMK Engineering College, Chennai
[2]Professor, Dept.of CSE, Velammal Engineering College, Chennai
knt.ece@rmkec.ac.in, emailpandiandurai@gmail.com

Abstract

Malicious traffic classification is the initial and primary step for any network-based security systems. This traffic classification systems include behavior-based anomaly detection system and Intrusion Detection System. Existing methods always relies on the conventional techniques and process the data in the fixed sequence, which may leads to performance issues. Furthermore, conventional techniques require proper annotation to process the volumetric data. Relying on the data annotation for efficient traffic classification may leads to network loops and bandwidth issues within the network. To address the above-mentioned issues, this paper presents a novel solution based on artificial intelligence perspective. The key idea of this paper is to propose a novel malicious classification system using Long Short-Term Memory (LSTM) model. To validate the efficiency of the proposed model, an experimental setup along with experimental validation is carried out. From the experimental results, it is proven that the proposed model is better in terms of accuracy, throughput when compared to the state-of-the-art models. Further, the accuracy of the proposed model outperforms the existing state of the art models with increase in 5% and overall 99.5% in accuracy.

Keywords: LSTM, traffic classification, artificial intelligence, malicious traffic.

Introduction

Intrusion Detection System (IDS) is widely used alert system to monitor computer and networks. IDS simply check the network traffic and user authentication level of security, if there any unnecessary activity occurred in the network the IDS will alert the user. This software used in various applications to defend the system from threats. The system monitoring the computer or network in continuous manner to identify any kind of violation in the system will report the result to administrator of the network or alert the centralized monitoring system such as Security Information and Event Management (SIEM) in the network. An SIEM uses the alarm filtering techniques to separate the false alarm and malicious activity. Honeypot is one of the IDS methods to customize detection rules and security policies with specific malicious thread detection. Some systems are used to detect the intrusion in the network but it not satisfies the expected result of system monitoring. From that aspect the organization concentrated on both prevention and detection. They uses Intrusion Detection and Prevention System (IDPS) to focusing the security policy of the system defined, network protocol rules and existing threads documentation. IDPS is the extended or up gradation of the IDS system, both are used to identify the malicious activity and intruders in the network. The main different is IDS can only detect the threads but in IDPS system can actively block or prevent the system from malicious threads which are detected. An Intrusion Prevention System (IPS) can take such actions in the network and sending alert message to computer. They also have permission to monitor the network

traffic, dropping of malicious packets in the network, block the access from unauthorized IP address, medicating TCP issues, and also check and correct the Circular Redundancy Check (CRC) errors. IDS can have various detection and classification methods. They are signature based, state full protocol analysis and anomaly based. The classifications of IDS are Network Intrusion Detection System (NIDS), Host Intrusion Detection System (HIDS), Protocol based Intrusion Detection System (PIDS), Application Protocol-based Intrusion Detection System (APIDS) and Hybrid Intrusion Detection system. To test the IDS system, the experts analyze the system with various attacking strategies to verify the efficiency of the newly invented or updated IDS system. They use various attacking method to verify the quality of the system. Such attacks are DDoS, Trojan horse attack, evasion of IDS attack, surveillance attack, and exploit attack. Anomaly based IDS is a recent trending method in IDS detection systems. This method uses the machine learning approach to detect the different type of malware in the network or computer. This system focuses on detecting the unknown attacks and overcome the traditional signature based IDS draw backs. This model also uses the defined pattern approach with the addition of machine learning method to create new defending patterns to detect the unknown malicious attacks and reporting them to the system admin. The above-mentioned methods suffer due to the detection rate and struck in false positive rate.

Contribution in this paper

- A novel malicious traffic classification system using LSTM model. The uniqueness of the paper is that the LSTM used in this paper completely replaces the traditional node-based classification error in the hidden layer of a network by introducing "memory cells", which overcomes the problems faced by the RNN architecture. Practically the LSTM architecture has shown improved results over the RNN architecture.
- The proposed traffic classification model works impressively well for any sort of sequence to sequence prediction, and by pairing it with LSTM, we can utilize the inherent nature of LSTM in recognizing the long-term dependencies of the sequences; hence next predicted traffic sequence would be fairly accurate.
- Consideration of multiple data sources to detect and distinguish the malware activities from user activities.

The organization of this paper is as follows that section 2 provides useful literature about the malwares and intrusion detection techniques, section 3 gives a clear view about proposed methodology. Section 4 describes about experimentation, section 5 follows with the result and performance analysis and section 6 concludes with impact of results and the future work

Literature Survey

This section explores the state of the art methodologies, techniques used in the current IDS for the malicious traffic classification.

A new IDS system was proposed by Zhan Xin [5] to detect the threads in WLAN networks. They introduced the system architecture to adopt the browser/server mode, and the system displays the result to client, it consists of the client and server interactions through the web browsers. The overall system architecture is focus on data storage layer, data acquisition layer, result analysis layer, and detection & analysis layer. They also introduced the block chain intrusion detection for more secure and reliable services. And also overcome the privacy challenges by using the block chain intrusion detection method, it results the more efficiently protect the system from malicious threads. Yi Yi Aung [6] presented the collaborative intrusion detection based on k-means methodology to improve the detection accuracy in intrusion detection system, they use the data mining in hybrid method and single method. It result comparatively reduced the time complicity of the system between the single method and data mining in hybrid method. The authors also describe a method called concept of projective adaptive resonance, which is used to particularly reduce the system model training time and maintaining the detection accuracy. This result the data mining algorithm role in IDS inters of time complexity.

A CNN based intrusion detection system was presented by Yihan Xiao [7]. The authors use the KDD-CUP99 dataset to compare the performance of the IDS by using the CNN. It results the CNN based IDS model provide the higher detection rate and reduce the false negative rate of the system. Kumar & Sharma [8]. proposed the IDS for signature and anomaly based methods. The authors describe the intrusion detection in the cloud computing environment and hybrid IDS algorithm for improves the detection rate in the private cloud environment. Mohammed hasan ali [9] proposed the model for apple based intrusion detection and validation. They use NSL-KDD data set. And compare this model with the Extreme Learning Machine (ELM) approach for IDS with the hybrid Particle Swarm Optimization (PSO) technique. It results the model PSO-ELM shows improve the accuracy in intrusion detection system. Li D [10] proposed the IDS for large networks. The authors develop two algorithms such as reduce & cluster, which is used to detect the intrusion and improve the filtering rate of the system. This algorithm to reduce the false alert rate and improves the analysis process.

An ID algorithm using AdaBoost technique was used by W. Hu[10] in decision stumps as weak classifiers. Their system performed better than other published results with a lower false alarm rate, a higher detection rate, and a computationally faster algorithm. However, the drawback is that it failed to adopt the incremental learning approach. Shengyi pan[11] proposed the automated approach for hybrid intrusion detection. Authors proposed algorithm detects the intrusion from the data logs, in the accuracy rate of 73% with related dataset. But this approach not suitable for larger data set, problem is capturing of log file in the system very difficult. Thae hurley [12] proposed the HMM based intrusion detection techniques for software defined network. The authors use the Android OS platform to detect the anomaly behavior in the android systems with the accuracy of 85 %. The only problem in this approach, it expands the feature vectors, and increase the maliciousness code in data sets.

A detailed discussion about the cyber security issues and discussions was carried out by Md zahangir [13]. Author includes the neuromorphic cognitive computing method for IDS network in cyber security using the deep learning. They use the NSL-KDD data set with vector factorization approach. It result the increase accuracy in the classification in rage of 81.31% to 90.12%. Deep learning method reduces the human effect in the task, and improves the performance of the system. M. A. Aydin[14] proposed the hybrid intrusion detection system with the snort method. They use the Packet Header Anomaly Detection (PHAD) and Network Traffic Anomaly Detection (NETAD), methodology to detect the intrusion detection with low false positive rate. They uses the DARPA 1998 dataset for detect the intruders in the network. And also describe the misuses of the hybrid ids in anomaly detection and signature based detection in the system. Safwan Mawlood Hussein[15] proposed the effectiveness of the hybrid ids with snort with native bayes network to improve the performance of the hybrid ids system. They used KDD cup 99 dataset for her research of intrusion detection. It results the average false alarm rate is improved and bayes networks gives the j48 graft response.

A novel methodology to detect mobile device attacks using the anomaly based IDs with machine learning classification was presented by Dimitrios Damopoulos [18]. The authors use the four machine learning algorithms to detect the anonyms attack in mobile devices. They use 4 algorithms such as Bayesion network, k-nearest neighbors, random forest and radial basis function algorithm. It results the high true positive 99.8%. Souparnika jayaprakash [16] proposed a system for data base intrusion detection using the octraplet and machine learning based on anomaly detection system. They create the architecture to implement the role based access control and implement new data structure is called octraplet, which is used to store the sql queries. This method is improves the performance of the system and detection rate. Dimitar Nikolov [17] proposed the recurrent neural network classifier for network intrusion detection based on short or long term memory units. This approach is mainly focus on HTTP server based intrusion detection.

An IoT IDS was proposed by Marin E Pamukov [19] to improve the detection rate and performance of the detection system in IoT devices. The authors use multiple negative selection algorithms to reduce the errors in intrusion detection and it can runs without input operators. It results the device detect the intrusion with 90% succession rate. Chung-ming [20] proposed a host based IDS by using the machine learning, which is inspired by adoptable agent based artificial intelligence. This approach is detects the malicious attacks from the system call and protect the system from host based intrusion attacks. It also shows the exchange of packets between the computers by detection signals. Ved prakash Mishra [25] proposed the simulator system for IDS to detect the DDoS attacks and alarm about the attack to the administrator. It approach uses the core of IDS and IPS to simulate the software in self-execution mode. And protect the system from traffic information. It results the system with increase in the performance of the detection system with factors of accuracy, security. This approach used for education purpose because of some implementation difficulty in the real time network devices.

A security system focusing on the IoT network devices was proposed by Hafeez [1] to detect the malicious activity in the IoT network devices. The authors use the IoT keeper method to detect and analysis the malicious activity in IoT devices. They use the C-means clustering and fuzzy interpolation algorithm to effectively detect the intrusion in the network. Han [2] proposed the system to detect the anomalous traffic in the network controller. The authors use the novel classifier technics to detect the intrusion in the controller network devices. They use cross entropy and SVM algorithms to detect traffic in the network. It improves the system detection rate and accuracy. Wei Wang [3] proposed the malware classification technic to analysis the network traffic and detect the malicious activity in the network. They use the CNN algorithm to train and test the data set for detect the malicious traffic in the network. It results the system with high detection accuracy. Sibi Chakkaravarthy [23-24] proposed the IDS to detect the intrusion in the wireless LAN networks. The authors [4] discuss the various causes of novel wireless intrusion attacks. They use combination KDE and HMM algorithms to detect intrusion in the network. It works through tandem queue feedback method. It detects the intrusion with the accuracy of 98%.

Proposed model – LSTM model

Long Short Term Memory (LSTM) is an advancement of Recurrent Neural Network (RNN). RNN is the first deep learning algorithm to retain the input state of users. The prediction using RNN is based on short term dependencies. While dealing with time series data short prediction based on short term dependencies does not give accurate results. A piece of information about the previous data might give a variant information in behavior analysis of a network. The minute information on the long-term dependencies may lead to a completely different and more accurate prediction. LSTM network prediction is based on both long term and short-term dependencies, which increases the prediction accuracy in time series data. STM network retains both input and output state of users. LSTM approach is highly adaptable for the analysis on timeseries data with time lags of unknown size.

LSTM Structure

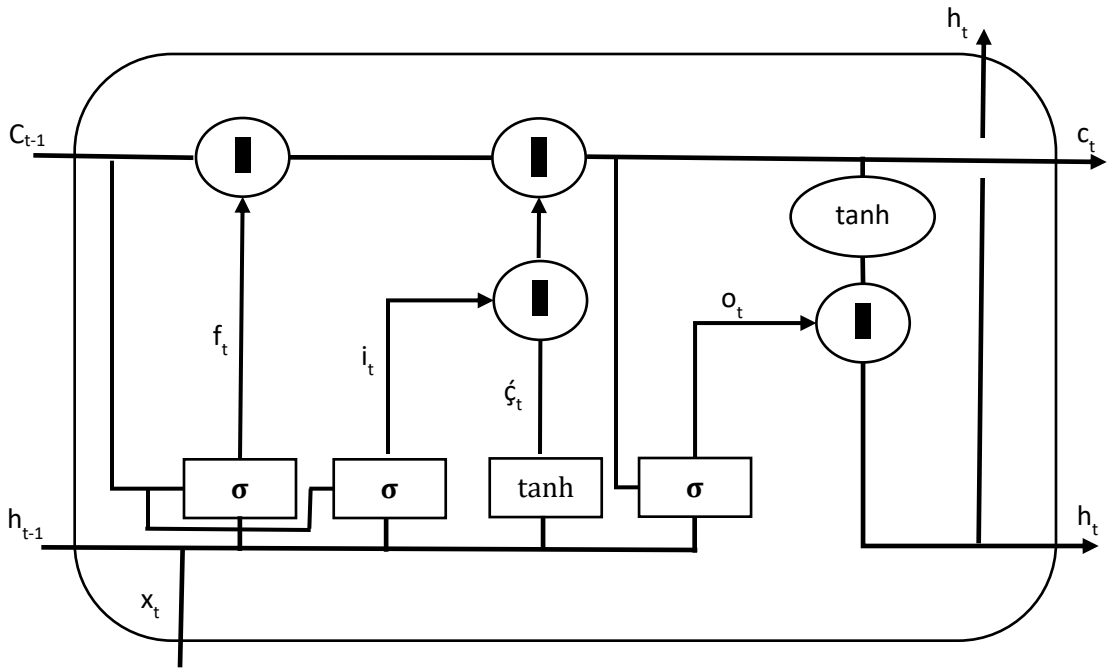

Figure 1: Structure of LSTM

LSTM is a three layer model with loop like structure which can add, delete and modify data easily. LSTM cells are build with input gate, output gate and forget gate as shown in figure 1. The sigmoid function is the activation function(α), it enhances the fitting ability of the model. All three layers contains activation function. The first layer contains forget gate, decides whether the information should be retained or deleted. The next layer has input gate and candidate state gate. The second layer decides which layer should be stored in a memory cell, whether the value should be stored in a new memory cell or updated in existing memory. The third layer has output gate, which controls the value stored in the cell to compute output activation. Based on all three layers the cell retaining, updating or deleting is done based on these three gates, it further composed of memory cells and multiplicative gates. The LSTM process with respect to the gates is detailed below

Forget Gate

The forget gate removes the information of no use. The current input x_t and previous output h_{t-1} are used in this layer. The two parameters are multiplied with weight and the bias is added to them. Activation function process the above result and gives the output in binary form. The output will be either 0 or 1. Where 0 represents that the information is forgotten or reset, and1 represents the information should be retained.

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \qquad (1)$$

Input Gate

Input gate uses the cell state retained as necessary information. The activation function is used for the regulating the information and the values to be retained are filtered. The input function is passed through a tanh function, which gives the output from -1 to +1. The vector values and the regulated values are multiplied with the weight to obtain the value of input gate.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \qquad (2)$$

Output gate

The output information is extracted from the current cell state in output gate. The tanh function is applied to the generated vector on the cell. As in input gate the information is regulated by the activation function and the value to be retained are filtered. The value of output gate is calculated using the below formula.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \qquad (3)$$

The result of the output gate is the binary output [0,1]. All the three gates using sigmoid activation function.

Candidate memory cell

Candidate memory cell computes in the same way as the above gates as shown in equation 4. The activation function tanh is used so the output varies from -1 to +1.

$$\acute{c}_t = tanh(W_c x_t + U_c h_t + b_c) \qquad (4)$$

Memory Cell

In LSTM to Compute c_t, the input gate and forget values are multiplied with the old contents in the memory cell. The value retained are used to perform this operation. The Current memory cell values are computed as given in equation 5.

$$c_t = f_t \circ c_{t-1} + i_t \circ \acute{c}_t \qquad (5)$$

For forget gate with the value 1 and the input gate value will be 0. The previous memory cell values c_{t-1} will be saved and used in current time whenever necessary.

Hidden States

Hidden state is computed using the output gate. The hidden state is calculated using equation 6. The output of hidden state varies from -1 to 1

$$h_t = o_t \circ tanh(c_t) \qquad (6)$$

When the output is 1 the memory information is efficiently passed to the predictor. For output 0 the information is retained within the memory cell. The expansion for the above used notations is described in table 1.

More hidden states can be added in the network but the increase in the hidden states does not increase the prediction accuracy.

Table 1: Table of notation

Notation	Expansion
x_t	LSTM units input vector
f_t	Activation vector of forget gate
i_t	Activation vector of input gate
o_t	Activation vector of output gate
h_t	Hidden state Vector
$ç_t$	Cell input activation vector
c_t	Cell state vector
$W_f, W_i, W_o, W_c, U_f, U_i, U_o, U_c$	Weight matrices for training
b_f, b_i, b_o, b_c	Bias for training

LSTM based network traffic analysis

A network is managed based on the prediction of real-time traffic volume. For accurate and efficient prediction long term dependencies play a vital role. LSTM extracts the temporal information from the traffic flow to analyze the behavior data of a network and predicts an application is malicious or not.Since network traffic contains time series data which are time variant and nonlinear. This increases the difficulty in the prediction of real time traffic, which leads to low accuracy problem. While the network traffic volume prediction problem is treated as regression problem, but it's a classification problem. LSTM is more suitable for network volume prediction problems. When dealing with massive data for network traffic prediction congestion control should be done for accurate predication of network behavior. For detecting the existence of malware, the opcode sequence is extracted and LSTM learns the features of malicious code sequence and pattern of network. If the opcode sequence of a file varies there exist a possibility of a file being modified dynamically by attacker. The malicious code is injected by the attacker in a normal file to launch an attack. When the malicious features or abnormal network behavior occurs from the LSTM prediction. LSTM analysis the network traffic and confirms the deviation in the normal traffic, a malware suspect is raised. The forget gate stores such abnormal dependencies for a very long time. Analyzing and detecting malware using LSTM is precise, since the LSTM is highly adaptable for networks with dynamic behavior. Here, the LSTM works as a classifier which differentiates the normal behavior with abnormal behavior and detects the existence of malware based on the increased abnormality.

Experimental Setup

The proposed LSTM model is experimentally validated using a real time testbed which consists of a wireless router, a laptop, attacker machine and an external packet capturing device. The external packet capturing device is attached to the laptop running Tshark utility to log all the tapped packets. The laptop is allowed to access the internet and connect with the local Wi-Fi network (established for experimentation).

Attacker machines are configured with Parrot Operating system and an automated python script is written to run the payloads required to launch the attack. Configuration for the above mentioned peripherals are given in the Table 2. Figure shows the illustration of the experimental setup.

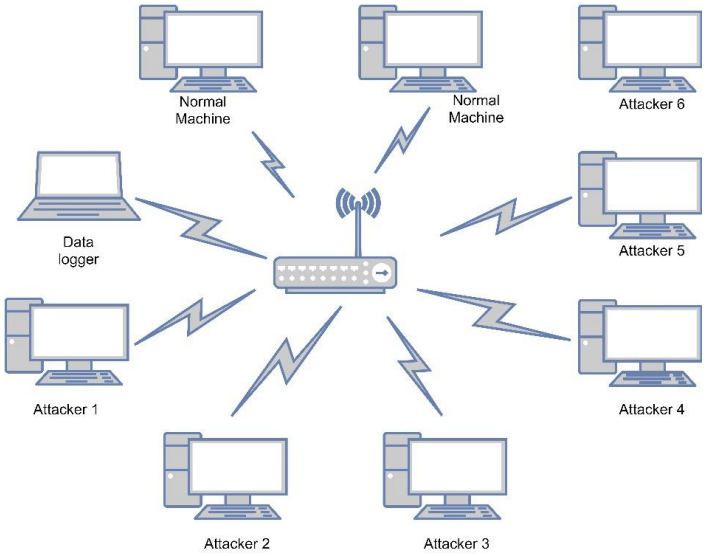

Figure 2. Testbed (setup) used for experimentation

Table 2. Device specification

S.No	Device name	Device Specification
	External packet capturing device	ASUS USB N13
	Laptop	4GB RAM, i5 processor, 8 GB RAM
	Attacker machine	8GB RAM, i5 processor, 8 GB RAM
	Wireless router	TP-LINK AC 1900, password protected using WPA2-PSK. Encryption: CCMP

Data set formation

The data set is collected from the internal network structure designed for the experimentation. The entire TCP flow is tapped and recorded. Every payload bytes of the TCP packet are recorded along with the TCP session. Each byte representation ranges from the value 0 to 255 in binary format. Then these bytes are normalized to a scale between [0,1]. The dataset collection takes nearly about 5 hours for collection with various range of scenarios such as malware network attack, normal network, DDoS attack etc. The attacks performed during the

data collection (given in Table 3) includes the malware traffic, authentication based attacks such as Fakeauth, Deauth, normal attacks such as SSL attacks, DNS attacks etc. The sequence for the payload is defined and ranges upto 1000 in the length. The total number of protocols monitored for recording is nearly 60. Python naïve data cleansing method is applied to remove the duplicates. Nearly 0.5 million records are removed after cleansing the duplicated records.

Table 3. Attacks and attack description.

S.No	Attack Prediction	Attack type and description
1.	Deauthentication	DDoS attack, Wide used to disconnect the AP and connected clients
2.	Evil twin	Wireless phishing attack which comprises of Deauth and Fake hotspot.
3.	Wifiphishing	Wireless phishing attack which comprises of Deauth and Fake hotspot.
4.	Caffe-latte, Chop-chop, Hirte	Attack which focuses on stealing Wi-Fi keys
5.	Arp replay (arp injection)	Used to generate IVs

Feature Learning

Feature learning always refer two important terms namely feature extraction and selection. The proposed LSTM model is used for the feature extraction and selection. This is performed by merging the node layers. The merging operations are handled by the deep layers which selects the information from the shallow layers. The information processed in the nodes except the outer layer are considered to be the features and the process is called as parameter tuning or learning. Furthermore, in the network input layers the nodes represent the features whereas in the hidden layers the features possess the activation property which is of deeper significance. The advantage of the proposed LSTM model is that the robustness involved in the dimensionality reduction. Since the data is of huge volume. The entire process is automated and the extracted features automatically gets mapped to the new feature space where the redundant information are filtered.

Feature Selection

Feature selection is performed based on the hyper parameter tuning. The entire LSTM model is trained as described in the above section. The stabilization of the proposed model is achieved by executing it with the collected dataset for multiple trails. Each trails records the error cost and epoch count. This means that the lower error cost within the minimum epochs. When the error costs around a very low i.e., 1/1000 and the epoch count which ranges from 80-100, this

trial can achieve an optimal hyper parameters. This modelled using the standard hypothesis as given below.

- Hypothesis1: The First two layers are considered to be the original features
- Hypothesis 2: Hidden layers always look for the features with appropriate weights.
- Hypothesis 3: All the absolute weights are summed with respect to the nodes.

Results and Discussion

Figure 3 shows the LSTM training model for malicious traffic classification.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Inputs (InputLayer)          (None, 150)               0
_________________________________________________________________
embedding_1 (Embedding)      (None, 150, 50)           50000
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                29440
_________________________________________________________________
FC1 (Dense)                  (None, 256)               16640
_________________________________________________________________
activation_1 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
out_layer (Dense)            (None, 1)                 257
_________________________________________________________________
activation_2 (Activation)    (None, 1)                 0
=================================================================
Total params: 96,337
Trainable params: 96,337
Non-trainable params: 0
_________________________________________________________________
```

The proposed LSTM structure defines the layers of the system, which can able to perform dataset CURD operations easily. Let us consider the scenario that the attacker can able to push the malicious traffic into the wireless network. Figure 4 shows the results obtained by tapping the traffic. From the Figures 4 and 5 it could be easily seen that the proposed LSTM based method is efficient in traffic classification. It is also observed that the traffic utilization carried out by the attacker are utilizing UDP based data transmission methods for exfiltration. Further these attacks targets the packet flow and always involves in the hijacking of the network.

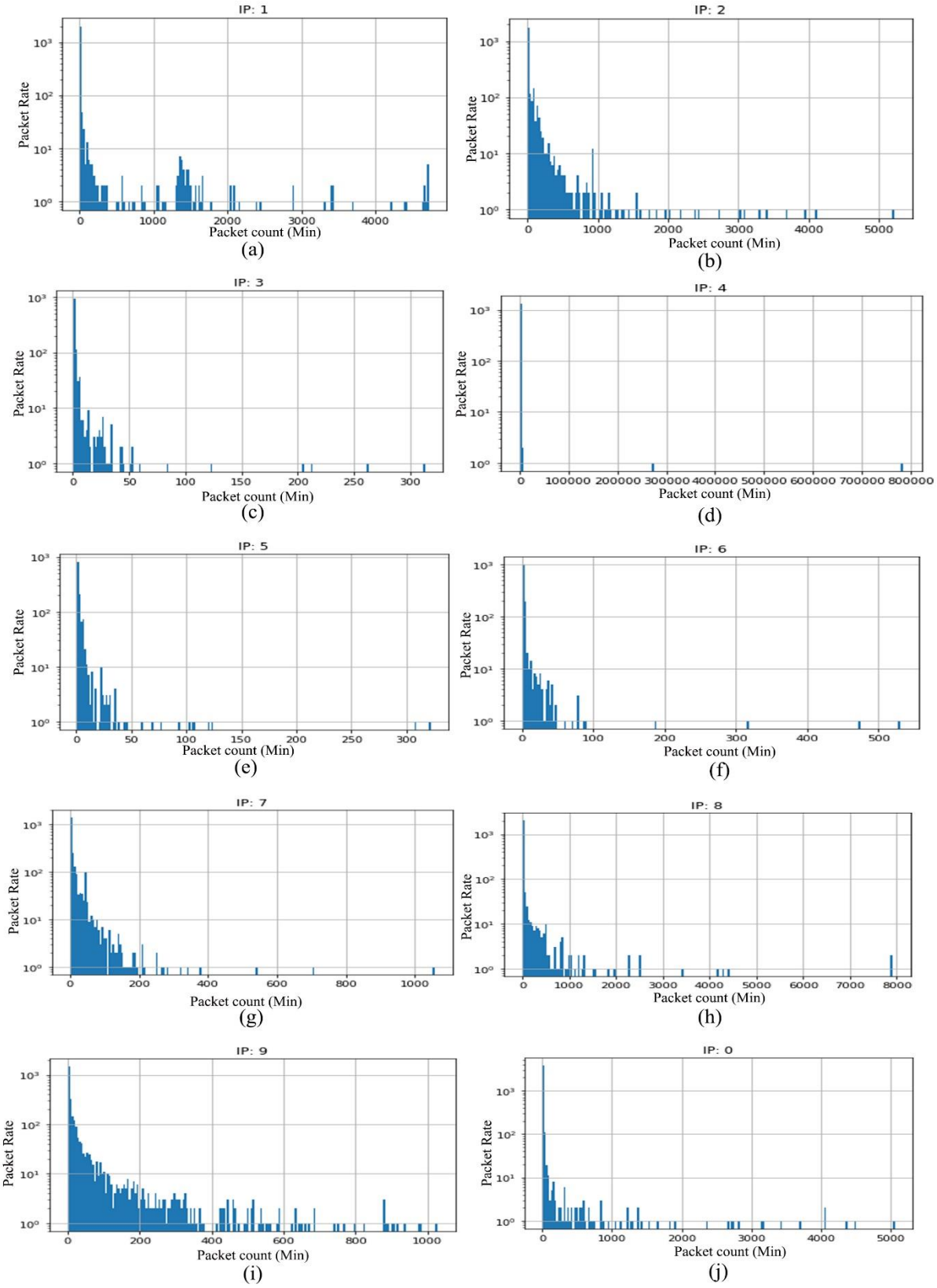

Figure 4. Results – Malware vs normal traffic.

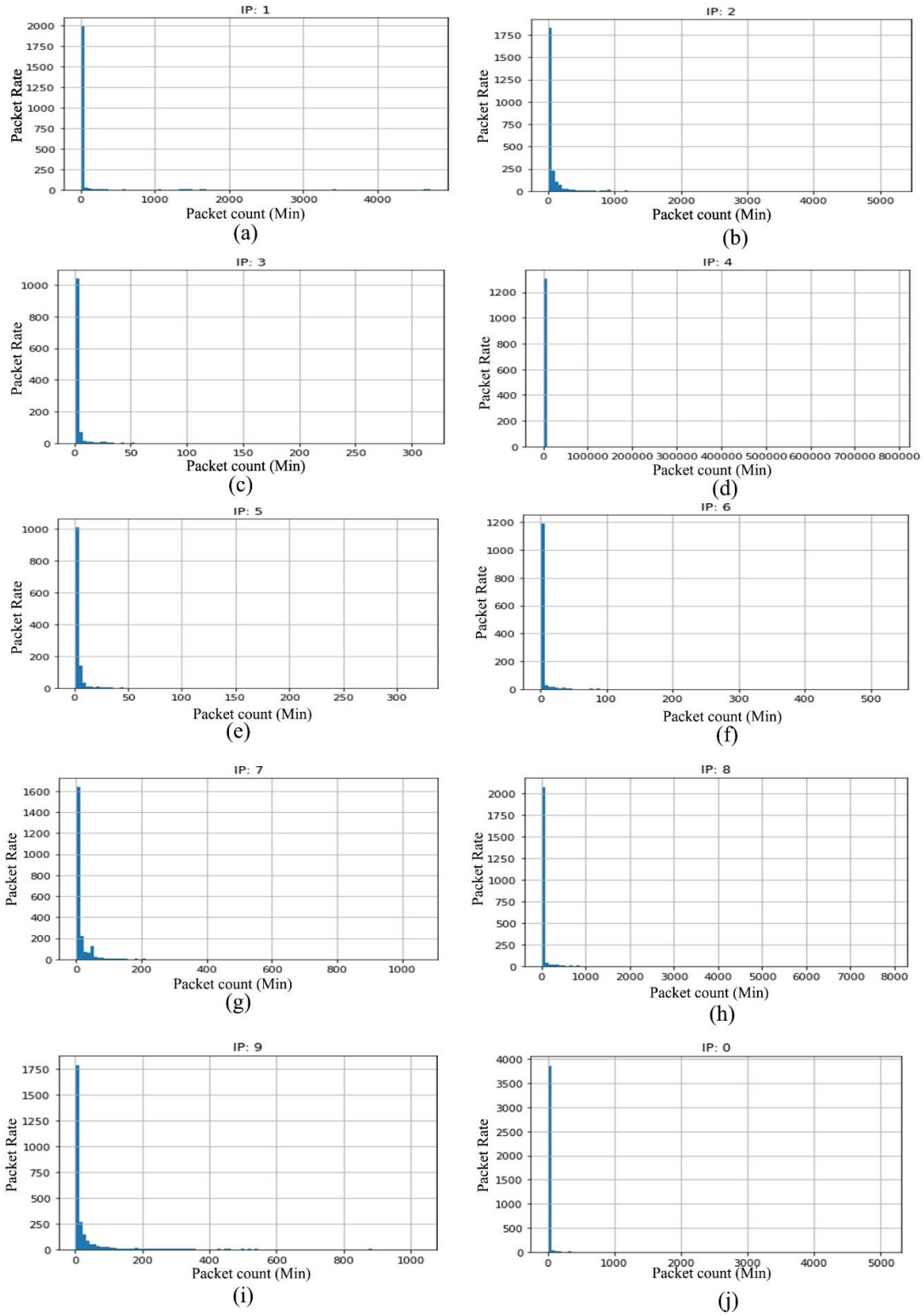

Figure 5. Results – Malicious DNS probes

Performance Analysis

The proposed model is tested in various environments with real time test bed. Assume that attacker configured with parrot OS with some python scripts to attack the system. The proposed model uses the feature extraction method to detect the pre-defined attacks and create the pattern for future possible attacks in an efficient way to detect the malicious attacks in wireless system. In this section the trained and tested data sets are plotted into the graph in various aspects. In execution phase each test trials records errors, it can be controlled with hypothesis values as listed in the LSTM. Figure 4 and 5 shows the variations in the network, if system analyze the 2 IPs and b shows the increase IP, likewise it estimates for the entire experimental setup as shown in the Figure 2. Figure 5b, 5c,5e,5g,5i shows the huge definition in the traffic when compared to the other figures (5a,5d,5f,5h,5j). From the Figure 4 and 5 it is clearly shown that the traffic generated by the attacker machine is very huge for a time period whereas the normal machine operates normally with the usual traffic. Figure 4d represents the normal traffic behaviour whereas the other Figures represents the abnormal traffic behaviour. Figure 6 shows the performance analysis of the proposed LSTM model with the state of the art models. From the performance results, it is clear that the proposed LSTM model outperforms all the state of the art models as listed.

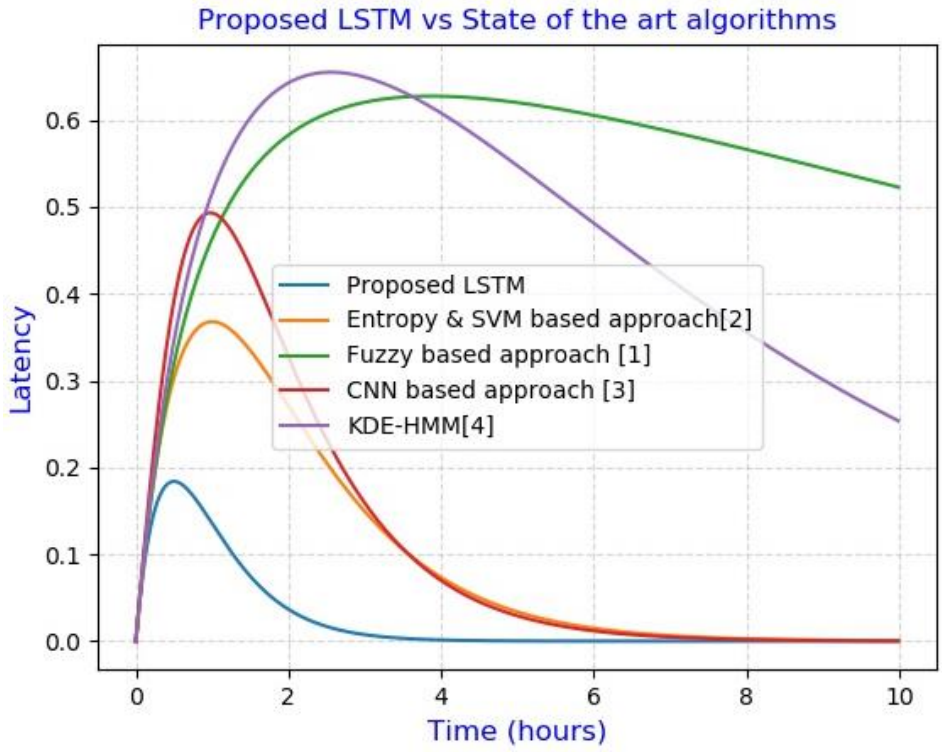

Figure 6. Performance metric – Latency (proposed model vs state of the art model)

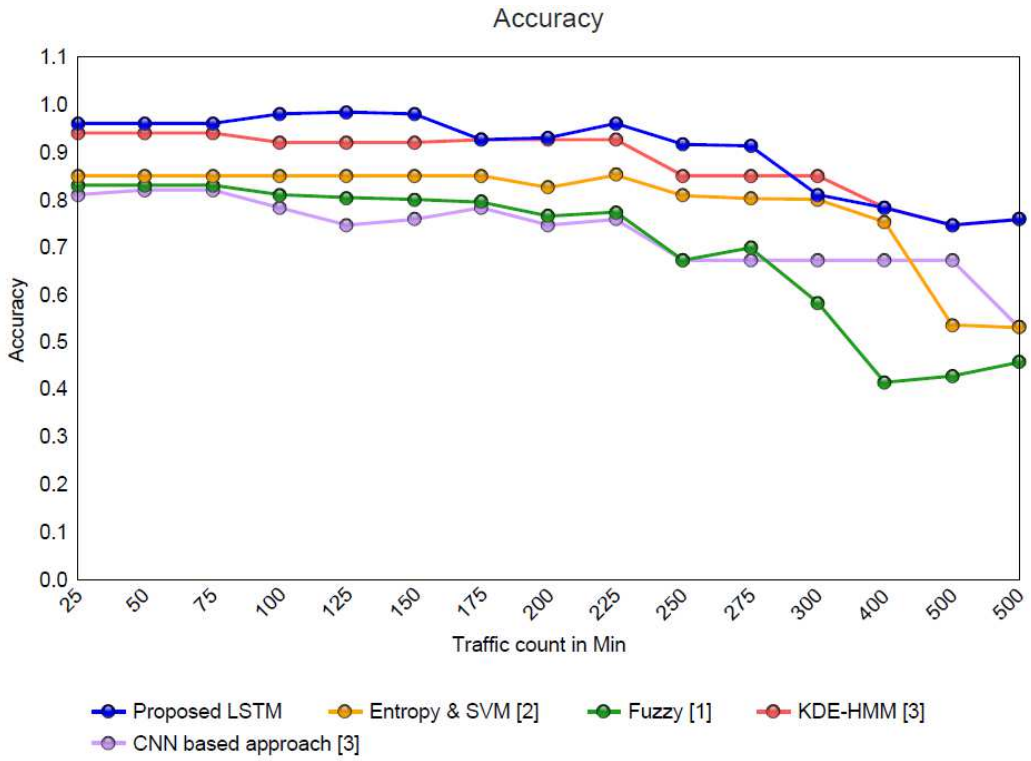

Figure 7. Accuracy of the proposed model (proposed model vs state of the art model)

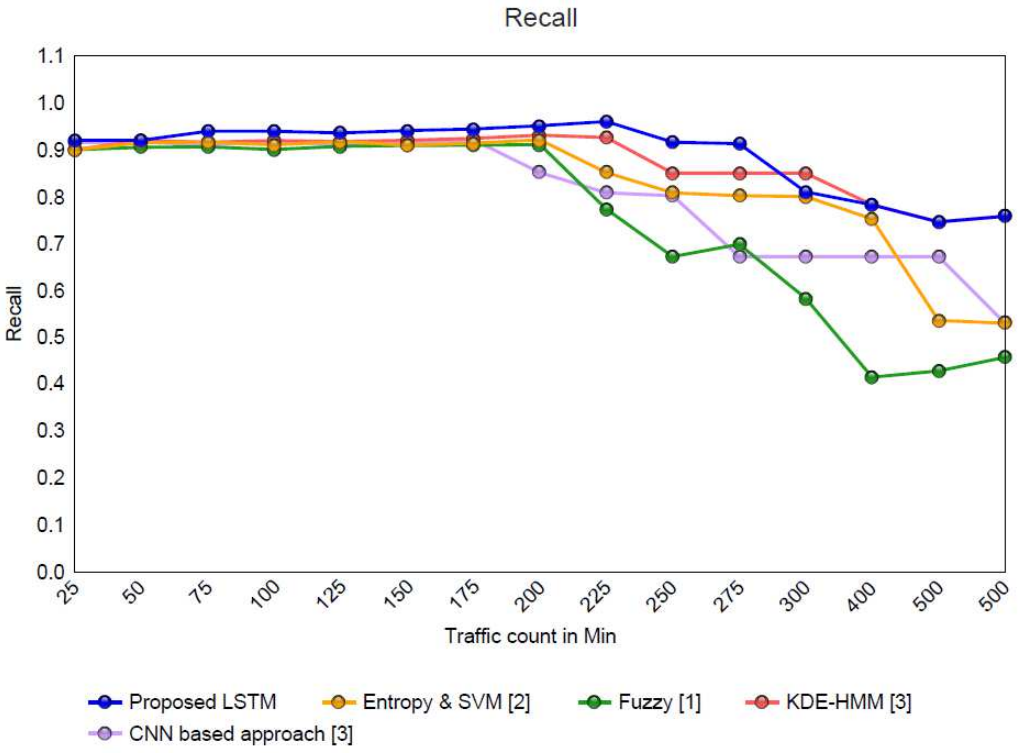

Figure 8. Performance metric – Recall (proposed model vs state of the art model)

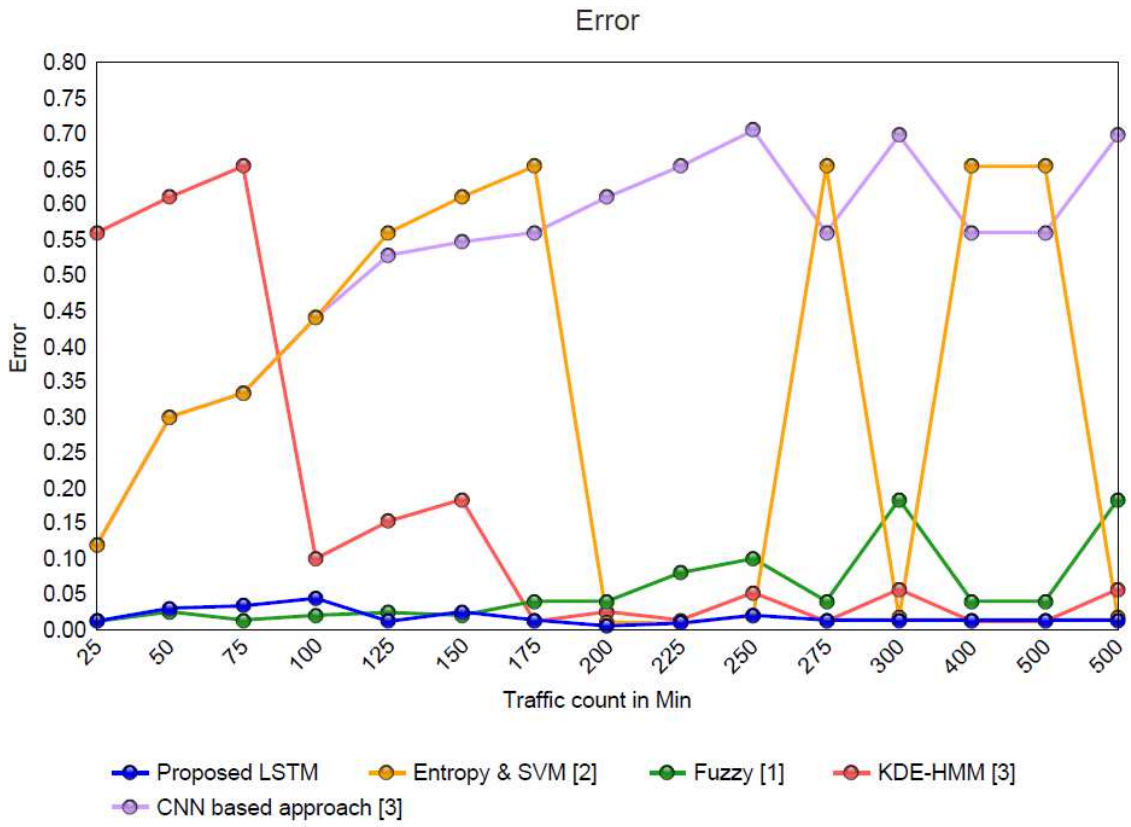

Figure 9. Performance metric – Error rate (proposed model vs state of the art model)

Conclusion

In this paper, a LSTM model was implemented with the combination of artificial intelligence sub base to analyze the network traffic. The existing model available in the literature detects the attacks based on the patterns and suffers in performance and scalability in the deployment. The proposed model helps to overcome the existing drawbacks and improvised the system performance in the accuracy at the rate of 99.5%. Figure 7 – 9 shows the performance metrics of the proposed model. From the results it is clearly proven that the proposed model outperforms the state of the art models available for traffic classification.

The comparison result of LSTM with the state of malicious traffic exhibits an outstanding performance with 99.5% accuracy in malicious traffic detection which also leads to attack detection.

The experimental results confirms that the proposed LSTM efficiently detects all the attacks which are listed in the papers [21-26]. The experimentation also confirms that there is a better increment inaccuracy, overall detection rate, reduced learning speed of the system. The proposed LSTM does not require any additional hardware, protocol modification, firmware upgrade in both the client and server. In future, this LSTM model can be extended to a mobile version. In addition, the metric selection for LSTM can be efficiently optimized using upcoming learning techniques.

References

[1] Hafeez, M. Antikainen, A. Y. Ding and S. Tarkoma, "IoT-KEEPER: Detecting Malicious IoT Network Activity Using Online Traffic Analysis at the Edge," in IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 45-59, March 2020, doi: 10.1109/TNSM.2020.2966951.

[2] W. Han, J. Xue and H. Yan, "Detecting anomalous traffic in the controlled network based on cross entropy and support vector machine," in IET Information Security, vol. 13, no. 2, pp. 109-116, 3 2019, doi: 10.1049/iet-ifs.2018.5186.

[3] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye and Yiqiang Sheng, "Malware traffic classification using convolutional neural network for representation learning," 2017 International Conference on Information Networking (ICOIN), Da Nang, 2017, pp. 712-717, doi: 10.1109/ICOIN.2017.7899588.

[4] S. C. Sethuraman, S. Dhamodaran and V. Vijayakumar, "Intrusion detection system for detecting wireless attacks in IEEE 802.11 networks," in IET Networks, vol. 8, no. 4, pp. 219-232, 7 2019, doi: 10.1049/iet-net.2018.5050.

[5] X. Zhan, H. Yuan and X. Wang, "Research on Block Chain Network Intrusion Detection System," 2019 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 2019, pp. 191-196, doi: 10.1109/ICCNEA.2019.00045.

[6] Y. Y. Aung and M. M. Min, "A collaborative intrusion detection based on K-means and projective adaptive resonance theory," 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Guilin, 2017, pp. 1575-1579, doi: 10.1109/FSKD.2017.8393000.

[7] Y. Xiao, C. Xing, T. Zhang and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks," in IEEE Access, vol. 7, pp. 42210-42219, 2019, doi: 10.1109/ACCESS.2019.2904620.

[8] R. Kumar and D. Sharma, "Signature-Anomaly Based Intrusion Detection Algorithm," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 2018, pp. 836-841, doi: 10.1109/ICECA.2018.8474781.

[9] M. H. Ali, M. Fadlizolkipi, A. Firdaus and N. Z. Khidzir, "A hybrid Particle swarm optimization -Extreme Learning Machine approach for Intrusion Detection System," 2018 IEEE Student Conference on Research and Development (SCOReD), Selangor, Malaysia, 2018, pp. 1-4, doi: 10.1109/SCORED.2018.8711287.

[10] W. Hu, W. Hu and S. Maybank, "AdaBoost-Based Algorithm for Network Intrusion Detection," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 38, no. 2, pp. 577-583, April 2008, doi: 10.1109/TSMCB.2007.914695.

[11] S. Pan, T. Morris and U. Adhikari, "Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems," in IEEE Transactions on Smart Grid, vol. 6, no. 6, pp. 3104-3113, Nov. 2015, doi: 10.1109/TSG.2015.2409775.

[12] T. Hurley, J. E. Perdomo and A. Perez-Pons, "HMM-Based Intrusion Detection System for Software Defined Networking," 2016 15th IEEE International Conference

on Machine Learning and Applications (ICMLA), Anaheim, CA, 2016, pp. 617-621, doi: 10.1109/ICMLA.2016.0108.

[13] M. Z. Alom and T. M. Taha, "Network intrusion detection for cyber security using unsupervised deep learning approaches," 2017 IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, 2017, pp. 63-69, doi: 10.1109/NAECON.2017.8268746.

[14] M. A. Aydin, T. Atmaca, O. C. Turna and H. Zaim, "Performance Study of New OBS Channel Scheduling Algorithms in a Multiservice Network," 2009 Fifth International Conference on Networking and Services, Valencia, 2009, pp. 242-248, doi: 10.1109/ICNS.2009.27.

[15] S. M. Hussein, F. H. M. Ali and Z. Kasiran, "Evaluation effectiveness of hybrid IDS using Snort with Naïve Bayes to detect attacks," 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP), Bangkok, 2012, pp. 256-260, doi: 10.1109/DICTAP.2012.6215386.

[16] S. Jayaprakash and K. Kandasamy, "Database Intrusion Detection System Using Octraplet and Machine Learning," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018, pp. 1413-1416, doi: 10.1109/ICICCT.2018.8473029.

[17] D. Nikolov, I. Kordev and S. Stefanova, "Concept for network intrusion detection system based on recurrent neural network classifier," 2018 IEEE XXVII International Scientific Conference Electronics - ET, Sozopol, 2018, pp. 1-4, doi: 10.1109/ET.2018.8549584.

[18] Damopoulos, D., Menesidou, S.A., Kambourakis, G., Papadaki, M., Clarke, N. and Gritzalis, S. (2012), Evaluation of anomaly-based IDS for mobile devices using machine learning classifiers. Security Comm. Networks, 5: 3-14. doi:10.1002/sec.341.

[19] M. E. Pamukov and V. K. Poulkov, "Multiple negative selection algorithm: Improving detection error rates in IoT intrusion detection systems," 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, 2017, pp. 543-547, doi: 10.1109/IDAACS.2017.8095140.

[20] C. Ou, "Host-based Intrusion Detection Systems Inspired by Machine Learning of Agent-Based Artificial Immune Systems," 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Sofia, Bulgaria, 2019, pp. 1-5, doi: 10.1109/INISTA.2019.8778269.

[21] D. Arivudainambi, K.A. Varun Kumar, S. Sibi Chakkaravarthy , P. Visu, “Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance”, Computer Communications, Vol.147, November, 2019, pp.50-57, Elsevier, (SCIE).

[22] Akshay T, S. Sibi Chakkaravarthy , D. Sangeetha, M. Venkata Rathnam, V. Vaidehi,“Role Based Policy to Maintain Privacy of Patient Health Records in

Cloud", Journal of Super Computing, Vol.75, Issue 9, June 2019, pp.5866–5881, Springer, (SCIE).

[23] A. Suresh, R. Kumar & R. Varatharajan, (2018) "Health Care Data Analysis using Evolutionary Algorithm" in Journal of Supercomputing. https://doi.org/10.1007/s11227-018-2302-0, 76, pages4262–4271(2020)

[24] S. Sibi Chakkaravarthy, D. Sangeetha and V. Vaidehi,"Intrusion Detection System to detect Wireless attacks in IEEE 802.11 networks", IET networks, July 2019, Volume 8, Issue 4, pp. 219- 232, IET.

[25] S. Sibi Chakkaravarthy, D. Sangeetha and V. Vaidehi,"A Survey on malware analysis and mitigation techniques", Computer Science Review, Vol. 32, pp 1 - 23, May 2019, Elsevier, (SCIE).

[26] V. P. Mishra and B. Shukla, "Development of simulator for intrusion detection system to detect and alarm the DDoS attacks," 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), Dubai, 2017, pp. 803-806, doi: 10.1109/ICTUS.2017.8286116.

[27] Suresh, A., Udendhran, R. & Balamurgan, M. "Hybridized neural network and decision tree based classifier for prognostic decision making in breast cancers" Soft Computing (2019). https://doi.org/10.1007/s00500-019-04066-4, 24, pages7947–7953(2020)

Figures

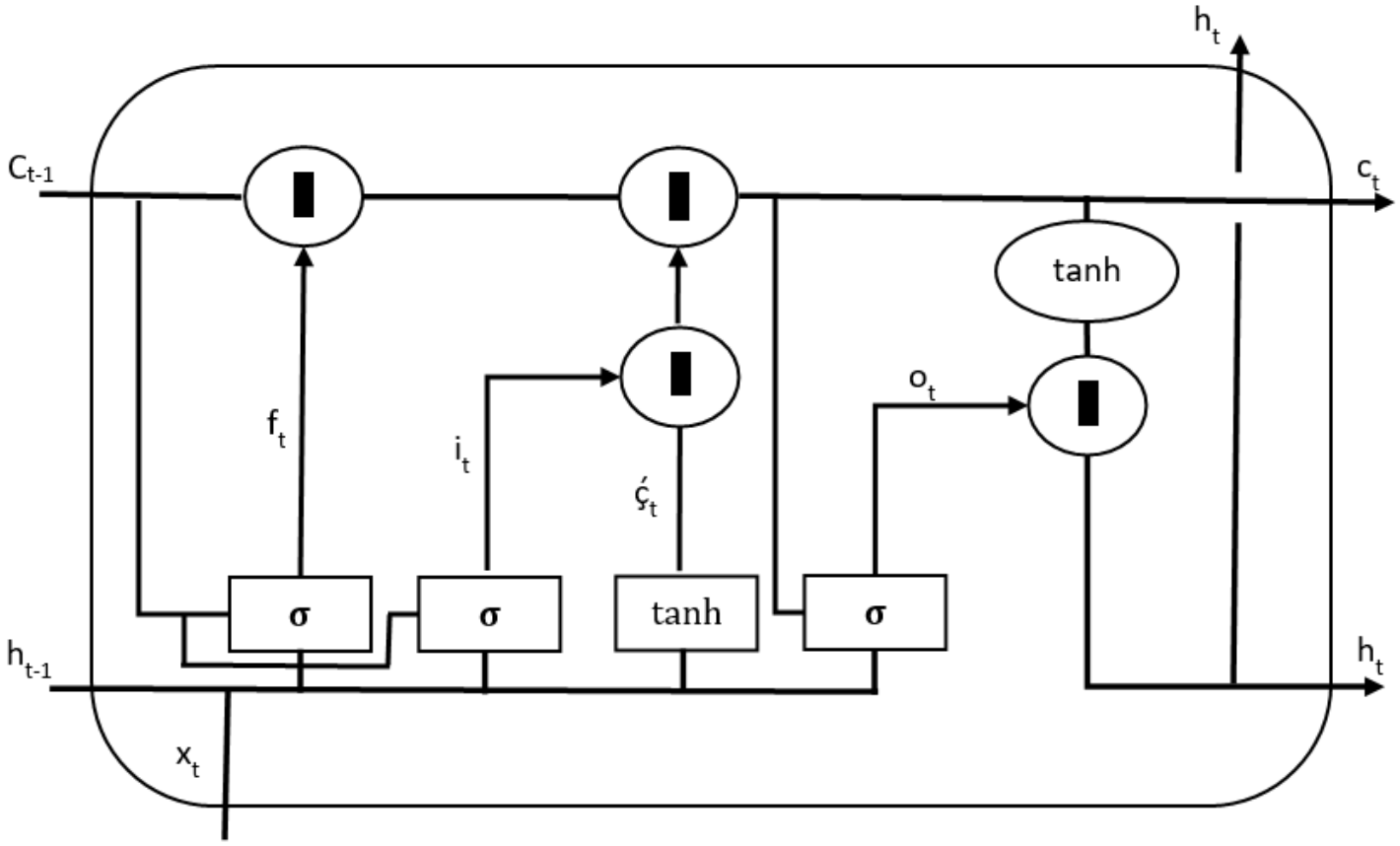

Figure 1

Structure of LSTM

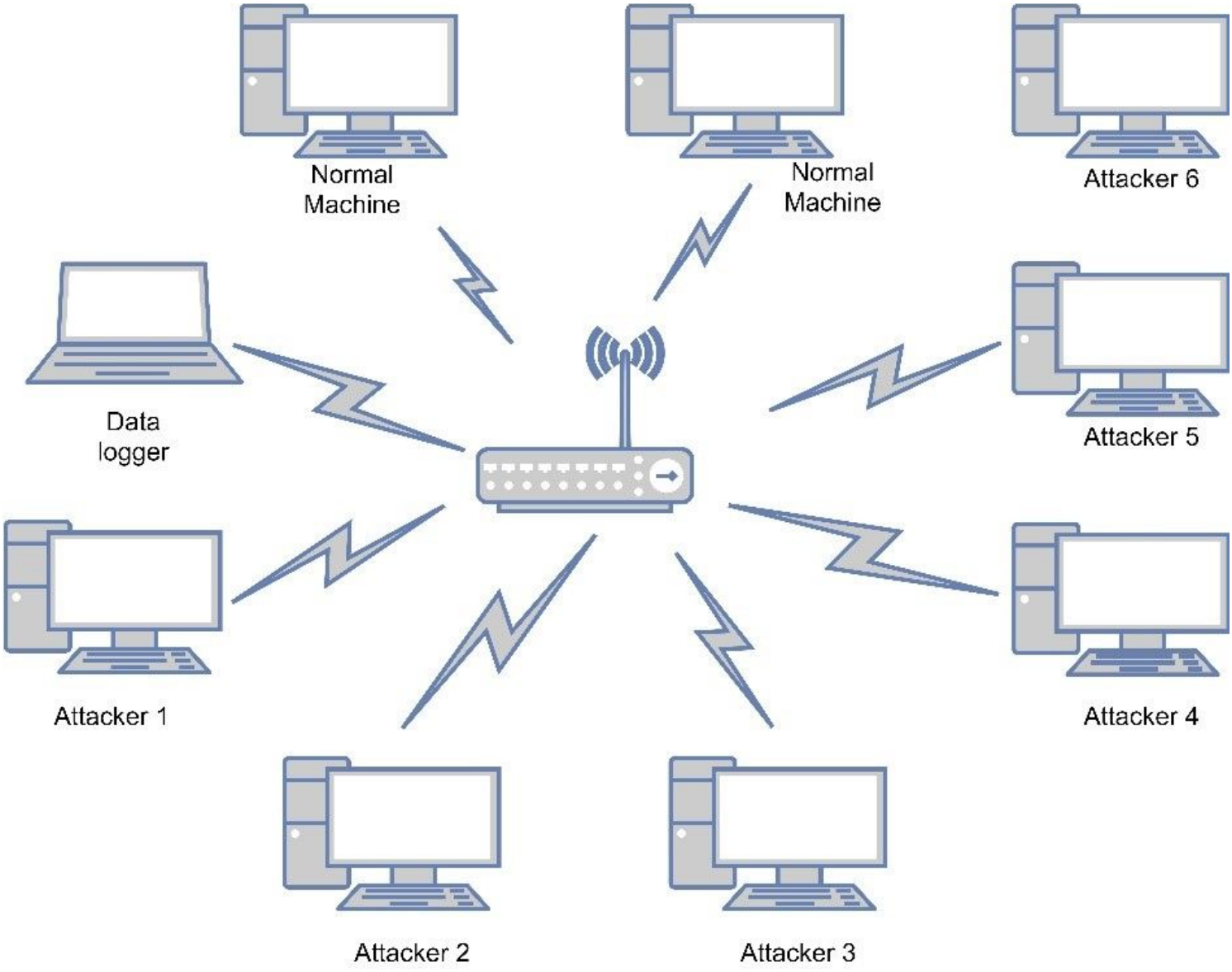

Figure 2

Testbed (setup) used for experimentation

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Inputs (InputLayer)          (None, 150)               0
_________________________________________________________________
embedding_1 (Embedding)      (None, 150, 50)           50000
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                29440
_________________________________________________________________
FC1 (Dense)                  (None, 256)               16640
_________________________________________________________________
activation_1 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
out_layer (Dense)            (None, 1)                 257
_________________________________________________________________
activation_2 (Activation)    (None, 1)                 0
=================================================================
Total params: 96,337
Trainable params: 96,337
Non-trainable params: 0
_________________________________________________________________
```

Figure 3

shows the LSTM training model for malicious traffic classification.

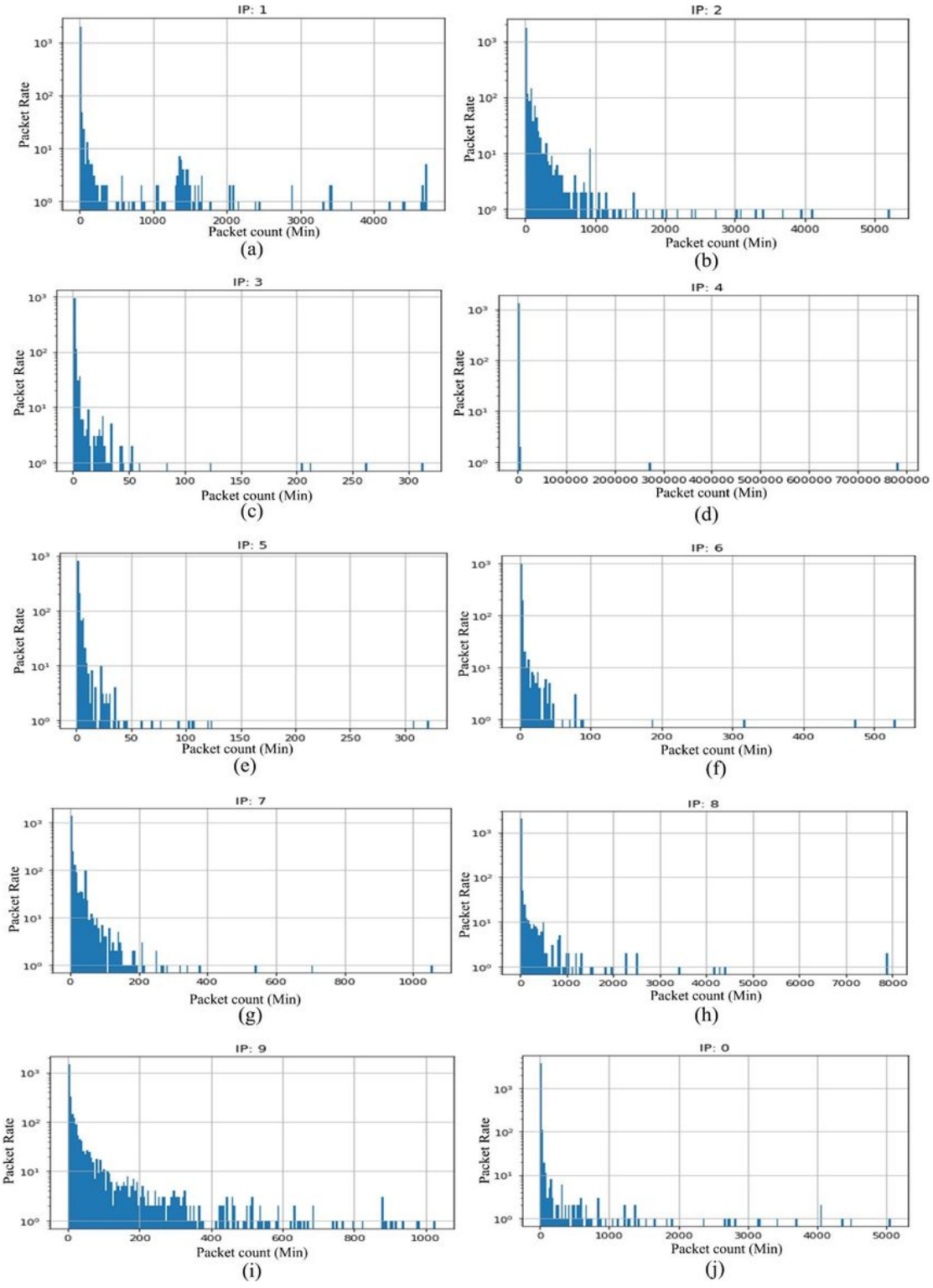

Figure 4

Results – Malware vs normal traffic.

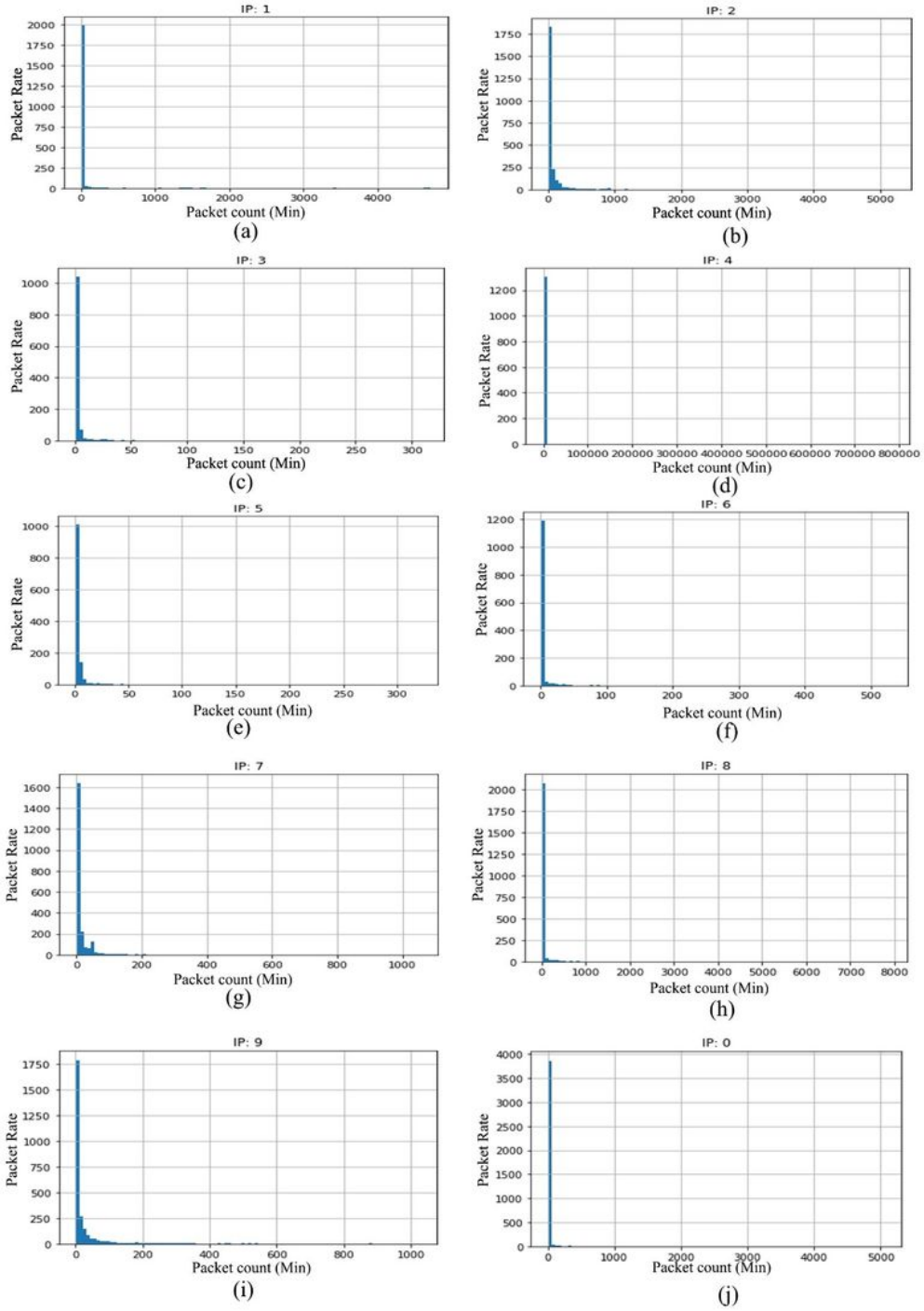

Figure 5

Results – Malicious DNS probes

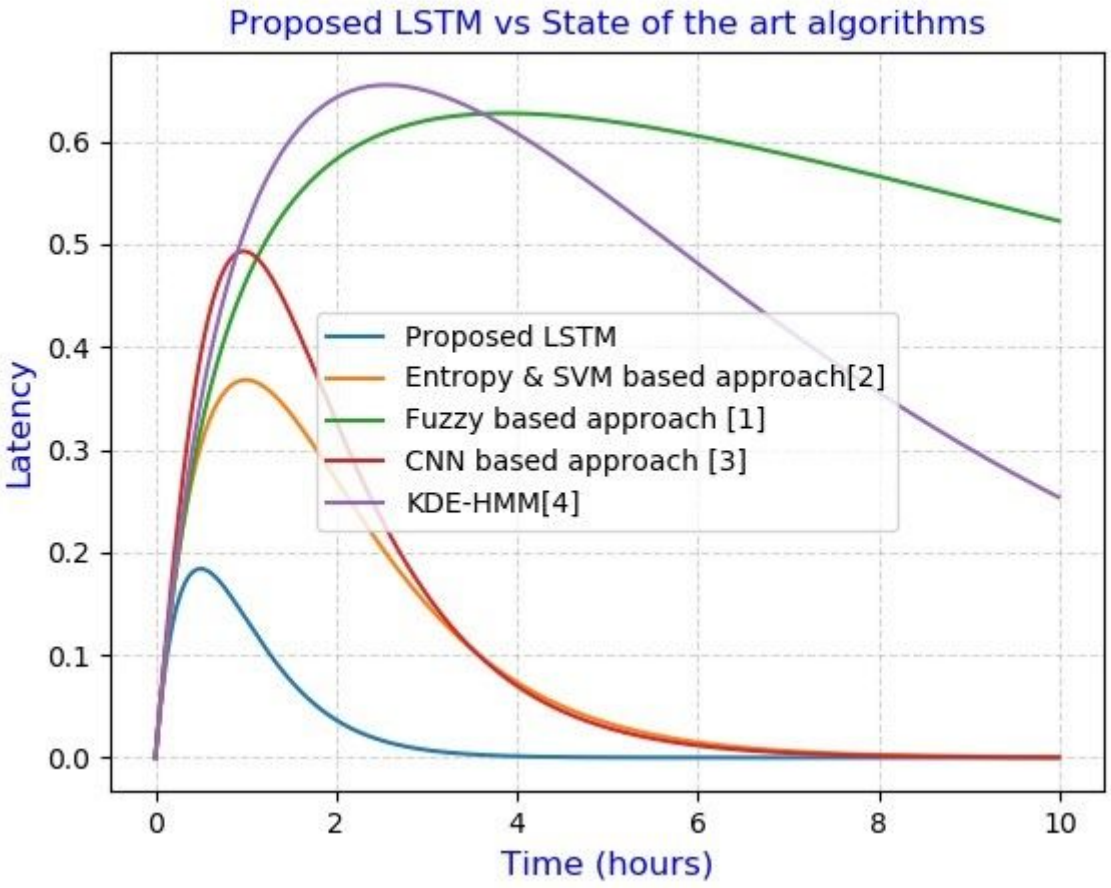

Figure 6

Performance metric – Latency (proposed model vs state of the art model)

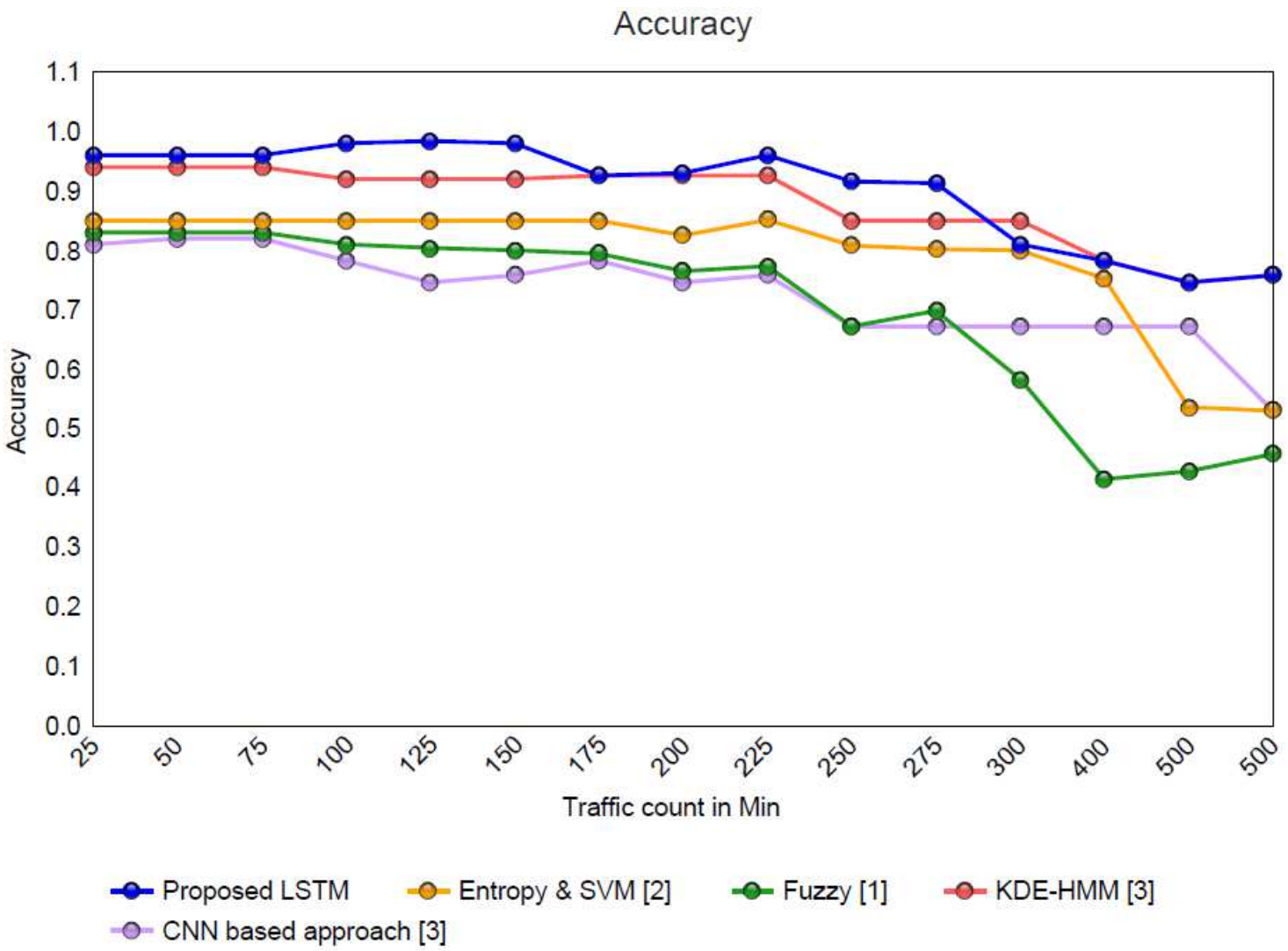

Figure 7

Accuracy of the proposed model (proposed model vs state of the art model)

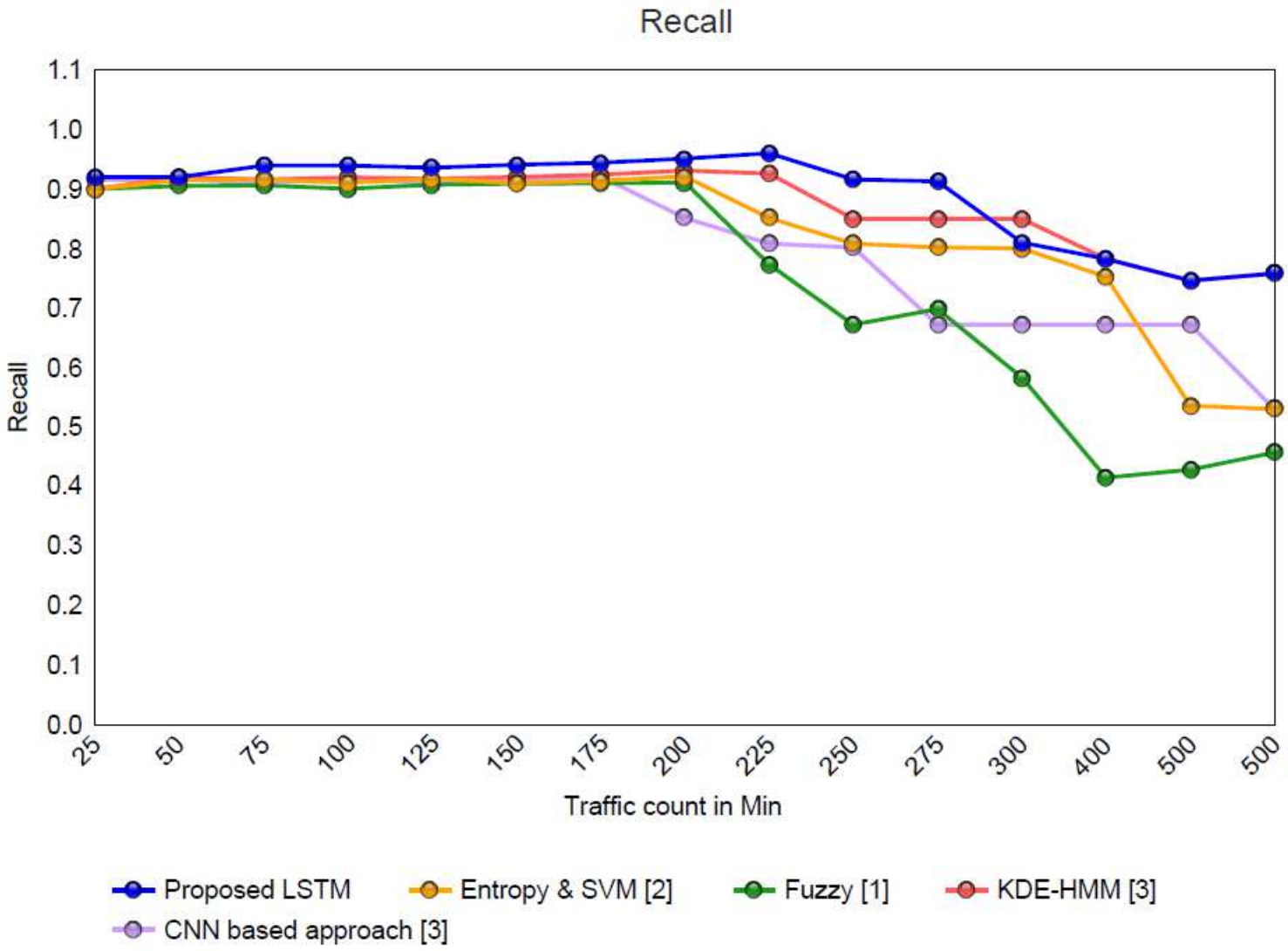

Figure 8

Performance metric – Recall (proposed model vs state of the art model)

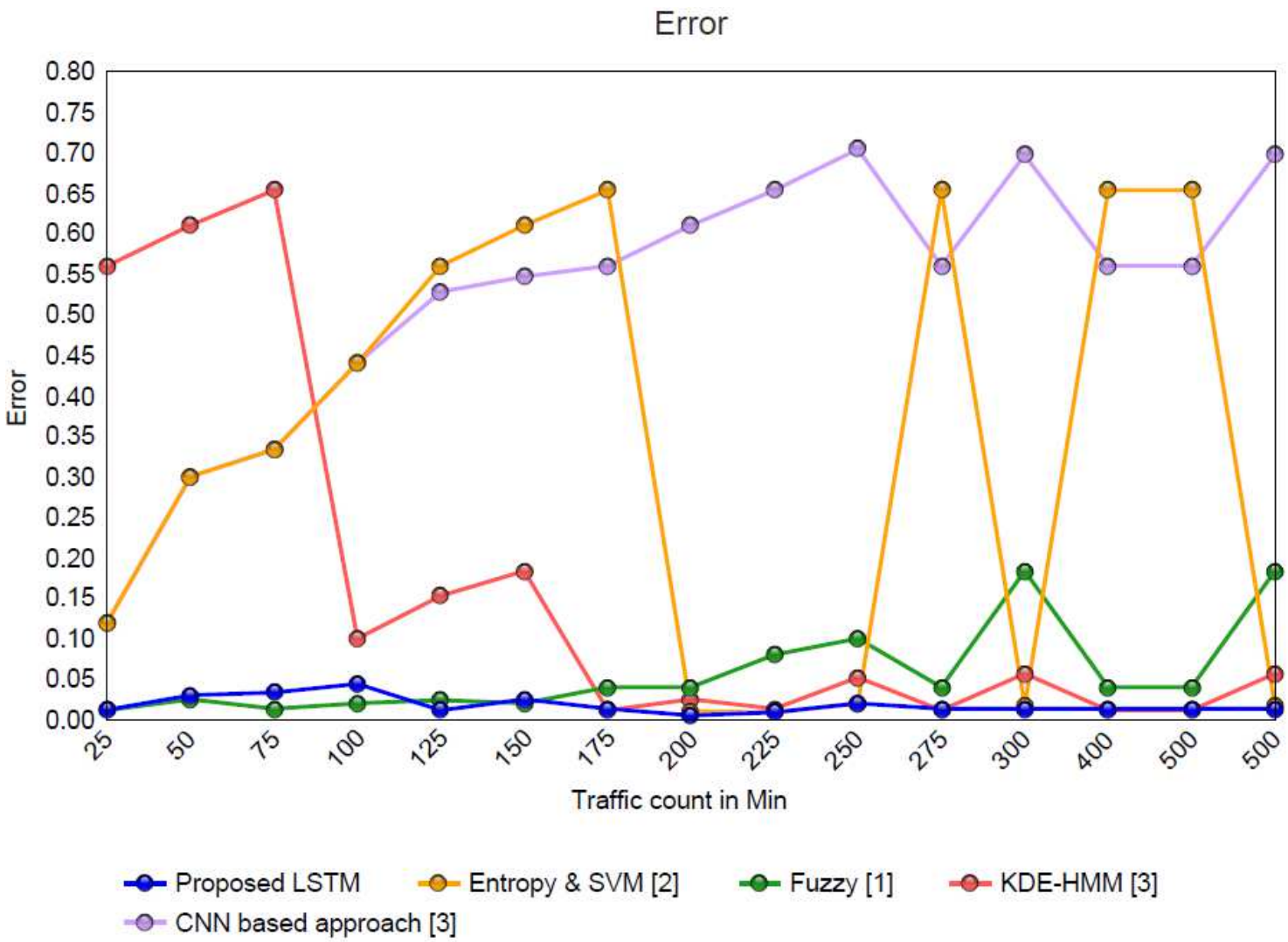

Figure 9

Performance metric – Error rate (proposed model vs state of the art model)