An Entropy-based DDoS attack Detection and Classification with Hierarchical Temporal Memory

Manh Hung Nguyen
Dept. of Electrical Engineering
Chung-Yuan Christian University
Chung-Li, Taiwan
Email: hungnguyen@cnsrl.cycu.edu.tw

Yu-Kuen Lai
Dept. of Electrical Engineering
Chung-Yuan Christian University
Chung-Li, Taiwan
Email: ylai@cnsrl.cycu.edu.tw

Kai-Po Chang
Dept. of Electrical Engineering
Chung-Yuan Christian University
Chung-Li, Taiwan
Email: kp.chang@cnsrl.cycu.edu.tw

***Abstract*—Detecting real-time DDoS attacks is a big challenge for network security. This paper proposes a hybrid machine learning model for the detection and classification of DDoS attacks. The system consists of a real-time detecting module capable of processing Entropy-based features. In addition, the classification module, based on the Hierarchical Temporal Memory and KNN classifier, is capable of mining changes in Entropy features for the classification of different types of DDoS attacks. Furthermore, it has the incremental learning capability to learn new traffic behavior and recognize new types of attacks. Finally, the simulation is conducted based on the CICDDoS 2019 dataset. As a result, the proposed system can successfully identify different types of attacks with high accuracy and precision.**

I. Introduction

In the paper, we propose a new approach for DDoS classification using entropy-based features and Hierarchical Temporal Memory (HTM) algorithm [1]. The new method can detect DDoS attacks in real-time, discriminate against different types of attacks, update new attack patterns, identify unknown attacks, and work with imperfect data. The purpose is to develop a better DDoS detector, which can solve some current problems when detecting and classifying different DDoS attacks. However, some difficulties can be encountered, including finding feature sets that can detect as many different types of DDoS attacks as possible. Moreover, detectors must work in high-speed networks and have a high detection rate but low false negatives and false positives. Finally, the model should have continuous learning and updates capabilities in real-time. The proposed approach uses Hierarchical Temporal Memory and machine learning algorithms to recognize patterns in stream data. Combined with the HTM modules in multi-layers, the approach can analyze complex data and build more robust models with higher accuracy, precision, and detection rates. We get remarkable results when classifying different types of DDoS attacks from regular traffic in our experiments. Furthermore, together with Entropy-based features, the HTM algorithm and multi-layer model presents the comprehensive ability to detect DDoS attacks.

The paper is organized as follows. Section II provides a general background of the HTM and Entropy methodologies. The related works are presented in Section III. In section IV, we describe the proposed System Architecture. Section V discusses the experiment detail and evaluation results. Finally, our conclusion and future works are presented in Section VI.

II. Background

A. Entropy based method

Shannon entropy present the uncertainty or randomness of a distribution. It can also monitor network traffic behavior for abnormal detection effectively [2]. Entropy-based features are also utilized together with machine learning methods for traffic analysis achieving significant results.

The Shannon Entropy is defined by the equation:

$$H = -\sum_{i=1}^{n} p_i log_2 p_i$$

Where n is the total number of distinct items, and p_i is the occurrence rate of the item i.

In network traffic monitor, n can be the number of distinct values of a network data field such as TCP port, IP address, Packet length. While p_i is the occurrence rate of a value in a time interval of a data field. For example, we can calculate the entropy of source IP address in a time interval by counting the occurrence numbers of each distinct source IP address, which can be read in packet's IP headers belong to the observed time interval, then calculate all corresponding occurrence rates p_i and apply the Shannon Entropy equation.

B. Hierarchical temporal memory Algorithm

HTM [1] imitates human brains to learn and recognize patterns. HTM model uses the simple Hebbian algorithm in the learning phase, allowing HTM to learn each input data record only once. Therefore, HTM is so suitable for processing online data streams. HTM can recognize patterns fastly by unsupervised method. HTM models are less affected by noise, can be trained quickly, with incremental learning. HTM can map infinite numbers of input patterns to finite numbers of Sparse Distributed Representations (SDR). Sparse Distributed Representations (SDR) are binary arrays, which have two essential characteristics. First, SDR can compare with other SDRs to recognize how they are similar, look like a human's memory. Secondly, SDRs are highly noise-resistant and can be sampled without the loss of much information. An HTM

model has three most important parts: Semantic Encoder, Spatial Pooler, and Temporal Memory.

Semantic Encoder: The encoder encodes the features vectors or different data types to binary vectors on the Input Space. The minimal element of Input Space is called cells. One type of Encoder is Scalar Encoder.

Fig. 1. The basic structure of the HTM model consists of Semantic Encoder, Spatial Pooler, and Temporal Memory. [1]

Spatial Pooler: Spatial Pooling converts binary vectors on the Input Space to sparse arrays. The properties of the spatial pooler allow HTM to maintain sparsity and overlap of Input Space. Thus, similar input data have high overlap, and different input data have low overlap.

Temporal Memory: This part is responsible for two important things, firstly, it learns and creates sequences of active mini-columns from spatial pooler over time. Secondly, it makes prediction about what pattern is coming next based on the temporal context of each input. In Temporal Memory, each mini-column has many cells, and each cell represents for a different temporal context.

Finally, we can separate HTM applications into two stages [1]. The first stage is the training phase; the HTM application learns all patterns in the dataset, creates invariant representations (SDRs), and saves them in memory. The second stage is the inference phase; the HTM application can use that memory to interpret new input patterns and predict the following pattern with continuous learning. After fully training, HTM can have all the invariant object representations in its world.

III. Related works

A. Entropy based method

Daneshgadeh *et al*. [3] proposed a method to detect DDoS attack and distinguish High rate, Low rate DDoS attack and Flash Event. The paper utilizes Shannon Entropy and machine learning algorithms to detect abnormal events. Mahalanobis Distance metric is used to distinguish High rate DDoS attack, Low rate DDoS attack and Flash Event. The work uses KOAD algorithm to classify abnormal and normal traffic in an unsupervised manner, so it doesn't require labeled data.

Koay *et al*. [4] proposed a method which uses entropy based features and multi-classifier to detect abnormal traffic events. The paper has experiments with two types of entropy including regular entropy and separation entropy. Separation entropy can give the variation of two distinct entropy-based features. The method can utilize the rich information of multiple entropy features to improve detection rate and reduce false alarm rate. The paper proposed a system called E3ML which can utilize rich information of multiple entropy features and 3 machine learning algorithms include Recurrent Neural Network, Multilayer Perceptron, and Alternating decision tree, to classify abnormal events.

Xinlei Ma *et al* [5] proposed a method to detect DDoS by analyzing relationship between source IPs, and destination IPs by chaos theory. The method will collect network traffic and calculate normalized entropy of source and destination IP address. The model use Lyapunov exponent to calculate a rate of separation between two related entropy series, and define threshold rate of separation to detect DDoS attack. The experiment shows that the rate of separation will change significantly when DDoS attack happens.

B. Machine learning-based method

Machine learning gives computers the ability to learn from data, explore hidden patterns and relationships to give predictions for new data. Supervised machine learning algorithms need labeled data, while unsupervised machine learning algorithms can describe data structure with unlabeled data. Input data for machine learning algorithms are features, and they should be chosen carefully to improve accuracy and reduce computation time. Feature selection is a necessary phase to analyze high dimensional and noisy data.

Ikram Sumaiya Thaseen *et al*. [6] use multi-class support vector machine and chi-square feature selection to decrease training and testing time and increase the accuracy of each type of classification. Random forest is more appropriate with a large data set than SVM or Naive Bayes, and also can adaptive with data size. However, Random forest takes a longer time for training, but less time for predicting. Random forest and decision tree can learn from data features, and define rules to separate dataset into many branches. Kamarularifin Abd Jalil *et al.* [7] compares the performance of Decision Tree, Support vector machine, and neural network.

Phurivit Sangkatsanee *et al*. [8] proposes a real-time IDS using Decision Tree. Nearest neighbor and logistic regression are famous regression algorithm to find the most similar training data with the observation. However, they are memory-intensive and may have poor performance with high dimension data.

Deep learning is suitable to model complex non-linear relationships by learning multiple levels of data representations that correspond to different levels of abstraction. [9] Deep learning can learn complex patterns with high dimension data, but it may be high misclassification. Fanzhi Meng *et al*. [10] compares the performance of LSTM with other machine learning algorithms including SVM when classifying attack and normal instances in NSL-KDD dataset. The result shows that LSTM has outperformed 99% detection rate and accuracy.

C. Survey of Features for DDoS detection

In our survey, researchers leverage source information is extracted from packet headers to create new features, describing the nature of attacks. Researchers used many different feature sets to detect DDoS attacks. Some features are easy to extract

from packet headers, but some features are so complicated to calculate in real-time.

Panida Khuphiran *et al*. [11] proposed two feature sets, including window-based and packet-based. The features are used to detect DDoS attacks in the 2009 DARPA Intrusion Detection dataset. Qin *et al*. [12] proposed a method using entropy-based features to model normal patterns by clustering algorithm. The method calculates five entropy-based features including source IP, destination IP, destination port, flow duration, packet size. The entropy of packet size uses five packet size-level to appropriate with a high-speed network.

Daneshgadeh *et al*. [3] proposed a hybrid method to discriminate between normal traffic, DDoS, and Flash Event. The authors use two types of feature vectors. One vectors consists of the time interval, destination IP, and source IP Entropy for the online machine learning-based method. The other consists of the time interval.

Eray Balkanli *et al*. [13] proposed two feature sets to detect DDoS Attacks in backscatter darknet traffic. The paper proved that their method can detect DDoS attacks without features related to IP addresses and port numbers.

Koay *et al*. [4] proposed a method to classify normal and attack traffic in a dataset that has different kinds of DDoS attacks. The method uses fifteen regular entropy-based features and five entropy variation features. The entropy variation features based on the variation of two distinct regular entropy-based features, generated using the variation of Lyapunov exponent separation method[5]. The author claims that their method can detect DDoS attacks effectively across datasets with different intensities.

In the survey, we notice that Entropy-based features are the most common features, which are used by Qin *et al*. [12], Daneshgadeh *et al*. [3], Mao *et al*. [14], Koay *et al*. [4] to detect different types of DDoS attacks in DARPA, or CAIDA dataset. In addition, entropy is a compact form to describe the changing feature's distribution, which is very important in network anomaly detection.

IV. System Architecture

A. Features Extraction

We extract entropy features from DDoS datasets and observe entropy-based sequences of network traffic. We notice that entropy features can be affected by DDoS attacks strongly, and discriminate between different types of attacks. For example, when we visualize the entropy of Source IP, Destination IP, Source Port, Destination Port, and Packet length in CICDDoS 2019 training dataset, we find out changing entropy as DDoS attacks happen. In addition, instead of using the entropy of packet size, other proposed features are found by using the distribution of packet size and mean packet size. Since it is not easy to calculate each distinct packet size in a high-speed network, we adopt the packet size in eight different levels.

Finally, we propose the set of feature vectors as input data. The feature vectors consist of eight features, including the entropy of TCP source, destination ports, and packet length. Moreover, the average packet length, total packet count, and the distinct number of TCP source and destination ports are also included. We believe that our selected features can be used to discriminate against many types of DDoS attacks in CICDDoS 2019 dataset [15].

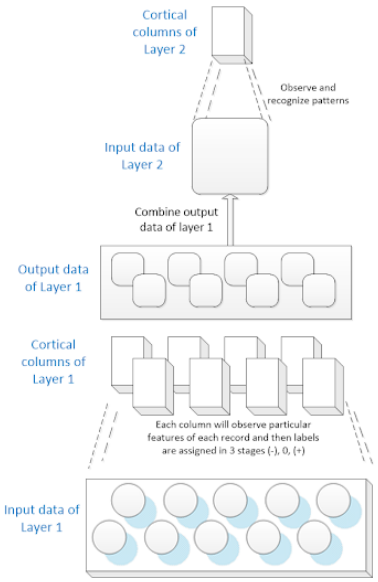

Fig. 2. The block diagrams of the proposed HTM-KNN model in two layers.

B. Signature of different type of DDoS attack

The proposed method aims to classify and recognize the DDoS attack signatures in an observation time interval of fifteen seconds. After extracting and analyzing features from the CICDDoS 2019 dataset, we notice that different types of DDoS attacks can make different changes in distributions of IP, port, packet size. We use Entropy-based features, numbers of distinct items in distribution, average packet length, and packet count to record the changes of network characteristics when an attack happens. By using this method, we can find out signatures of different types of attacks, and which types of attacks have the same signatures.

Table I shows how each features changes when a DDoS attack happens in CICDDoS 2019 dataset. The symbol of (0) means the attack has a feature value similar to regular traffic. The symbol of (-) represents the attack makes the feature value decreasing. Furthermore, the symbol of (+) means the attack makes the feature value increasing. The symbol of (--) represents decreasing more than symbol of (-), and similarly, the sign of (++) means increasing more than symbol of (+). Thus, as shown in Table I, we can easily find the different types of attacks based on the changing trend of features. For example, when comparing SYN(10) and TFTP(11) attacks, we can see that the Protocol Entropy feature is different. SYN(10) makes the feature value decrease, but TFPT(11) increases the feature value.

With the currently used features, we can discriminate between nine types of DDoS attacks. However, we still can not

distinguish between the attacks of SSDP(6), UDP(7), UDP-Lag(8), and WebDDoS(9) from the regular traffic.

C. Recognize pattern with HTM Cortical Column

1) Create SDRs from sequences of input data by HTM algorithm: Each input data may have one or combined many features. The scalar encoder is used to convert input data into binary vectors. These binary vectors are sent to the Spatial Pooler module to create SP-SDR and then send to the Temporal Memory module continuously to create TM-SDR. TM-SDR are outputs of the Temporal Memory module. They are SDR binary arrays and used as patterns to present a sequence of input data at a time interval. Each pattern represents the corresponding input data and its context. The final step is to use classification techniques to label prototype patterns in the training phase and prediction phase.

2) Find prototype patterns from SDRs in the training phase: In this step, we will use a clustering technique to the find prototype patterns from those created from training datasets. Prototype patterns are SDR binary arrays and can affect the result of the KNN classifier, which is used to assign labels for observed patterns.

The raw overlap is the method used to calculate the distance between two binary patterns. The distance value is the number of bits that differ between two binary arrays. The smaller the distance between two patterns is, the more similar they are.

We define the distance_threshold as the minimum distance between two prototype patterns. We define the distance_threshold as the minimum distance between two prototype patterns. Distance_threshold is an optimized parameter of the HTM-KNN model. Lower distance_threshold will create more prototype patterns in the training phase, and higher distance_threshold will create fewer prototype patterns. We need an optimized distance_threshold to create enough prototype patterns, and each prototype pattern is present for a variant of a type of attack. In the training phase, when HTM calculates a training pattern from input data; the model finds the smallest distance between the new pattern and all existing prototype patterns. If the smallest distance is higher than the distance threshold, the training pattern will be assigned as new prototype patterns. If the smallest distance is lower than the distance threshold, the new pattern will be assigned as absorbed patterns. We will assign labels for all prototype patterns, these labels are as same as those in the corresponding input data.

3) KNN assign labels for observed pattern in the prediction phase: KNN algorithm is adopted to assign the label for observed patterns in the prediction phase. In order to assign a label for input data, the model must convert the observed input data to an SDR binary array, also called an observed pattern. Then, the model can compare the distance between the observed pattern with all existing prototype patterns specified in the training phase and find k nearest prototype patterns. Finally, the observed pattern is assigned a label by major voting between its k nearest prototype patterns.

D. Two-layers HTM-KNN model

Using the HTM Cortical Column described in Section-IV-C, we can recognize the most similar attack signature based on the increasing and decreasing tendency of the features. We build a two-layered HTM model, as shown in Figure 2. to observe all network features and then recognize attack signatures to assign labels for observed patterns in the network traffic.

Layer 1 is responsible for the role of observing and assigning labels for all features for all features. Each HTM Cortical Column observes a particular feature using only three types of (-), (0), (+) labels. Label (0) means the attack makes the feature value similar to that of regular traffic. Label (-), and (+) represent the decreasing and increasing tendency of the feature value. Input data of layer 1 is a serial of records. Each record represent for each time interval and has values of a feature set.

Layer 2 is responsible for recognizing the matching attack signatures or most similar attack signatures and then predicting the type of attack. All output data from layer one are combined to become input data of layer two. In the layer two, the HTM Cortical Column converts the input data to SDR binary and compares the observed SDR with the most similar signatures. Based on that information, we can predict the type of attack against regular traffic. The two-layers HTM model can also remember the signature of different types of attacks in the training phase and then recognize attack signatures in the prediction phase.

V. Experiment and Evaluation

A. Simultion and testing environment

In our experiment, we use the merged CICDDoS 2019 dataset and MAWI dataset [16]. CICDDoS 2019 provides pcap files for benign traffic and the most updated common DDoS attacks. The attack flows are labeled by timestamp. CICDDoS2019, have one training dataset and one testing dataset. The training dataset has twelve DDoS attack types including NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP. The testing dataset has seven DDoS attack types including PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag and SYN. MAWI070201 dataset has a role of background traffic.

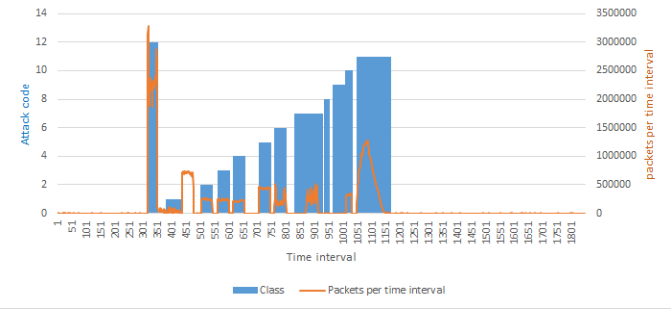

Fig. 3. Attack class and attack volume in Training dataset

TABLE I
THE TREND FOR ALL THE FEATURES OBSERVED DURING THE ATTACKING PHASE IN CICDDOS 2019 DATASET.

	IP Entropy		IP Distinct		Port Entropy		Port Distinct						Signature #
Attack type/Feature	Src IP Entropy	Dst IP Entropy	Src IP Distinct	Des IP Distinct	Src Port Entropy	Dst Port Entropy	Src Port Distinct	Dst Port Distinct	Pkt Len Entropy	Average Pkt Len	Pkt Count	Protocol Entropy	
NTP (12)	-	-	0	0	+	+	0	+	-	-	+	-	1
DNS (1)	-	-	0	0	-	+	0	0	0	0	0	+	2
LDAP (2)	-	-	0	0	+	+	0	+	+	+	+	+	3
MSSQL (3)	-	-	0	0	+	+	+	+	+	0	+	+	4
NetBIOS (4)	-	-	0	0	+	+	0	+	-	-	+	+	5
SNMP (5)	--	--	0	0	+	+	0	+	+	+	+	+	3
SSDP (6)	-	-	0	0	+	+	+	+	+	-	+	+	7-1
UDP (7)	--	--	0	0	+	+	+	+	+	-	+	+	7-1
UDP (7)	--	--	0	0	+	+	+	+	0	-	+	+	7-2
UDP-Lag (8)	0	0	0	0	0	0	0	0	0	0	0	0	0
WebDDoS (9)	0	0	0	0	0	0	0	0	0	0	0	0	0
SYN (10)	-	-	0	0	+	+	+	+	-	-	+	-	8
TFTP (11)	-	-	0	0	+	+	+	+	-	-	+	+	9
Portmap (13)	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE II
EACH NAME OF DDOS ATTACK WILL BE REPLACED BY A NUMBER OF ATTACK CODE

Types of attack	DNS	LDAP	MS-SQL	Net-BIOS	SNMP	SSDP	UDP	UDP-Lag	Web-DDoS	SYN	TFTP	NTP	Portmap
Attack code	1	2	3	4	5	6	7	8	9	10	11	12	13

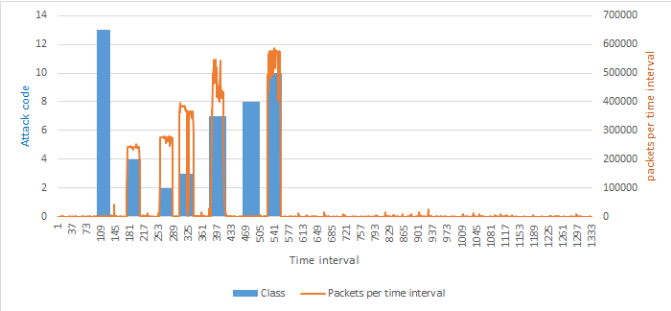

Fig. 4. Attack class and attack volume in Testing dataset

We get the PCAP files of the training and testing dataset to extract the features. In the experiment, we use eight features, including the entropy values of TCP source and destination port, entropy of packet length, the distinct number of TCP source and destination port, the average packet count and total packet count.

We create feature vectors for each 15 seconds time interval of the two datasets. Figure 3 shows the time intervals in DDoS attacks and the total packet of each time interval. We can notice that LDAP, MSSQL, NetBIOS, SNMP, SSDP, SYN DDoS attacks appear with high volume traffic. While UDP-Lag and Web DDoS appear with low volume traffic, UDP attacks appear with both low volume and high volume traffic. The model detect LDAP, MSSQL, NetBIOS, UDP, UDP Lag, SYN in the testing dataset. Figure 4 show that the attack volume of LDAP, MSSQL, NetBIOS, UDP, UDP Lag, SYN attack in the testing dataset has a similar attack volume with the same type of attack in the training dataset. Table II shows the attack code used in Figure 3 and Figure 4. Again, the attack code replaces the name of the DDoS attack.

B. Results and evaluation

The model learns all input feature vectors in the training phase, creates corresponding SDRs, chooses prototype patterns, and saves all prototype patterns in memory. Prototype patterns represent normal behaviors and all twelve types of DDoS attacks, including NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP.

The model converts the input data sequence in each observation time interval to corresponding SDRs in the testing phase. Then, KNN algorithm assigns labels for each observed SDR by analyzing the distance between the corresponding SDR with all prototype patterns detected.

As shown in the Figure 2, there are eight HTM cortical columns in the layer one. Therefore, for all observed features, labels of (-), (0), and (+) are assigned to the eight cortical columns for each time interval. Layer one signals layer two the observed signature of the current time interval. Then, Layer two continues to create SDRs representing the attack signatures from layer on and compare the observed SDR with all learned attack signatures in the training phase. Finally, the matching can be found with the most similar attack signatures.

In the testing dataset, there are seven different types of DDoS attacks. The model learned six types of attack patterns, including NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, and SYN DDoS attacks in the training phase. However, the model misses the PortMap attacks. It should be labeled as abnormal, and the model should consider Portmap as an unknown attack.

Table III presents the performance of HTM-KNN by metrics of detection rate, accuracy, and precision. The model achieves high detection accuracy and precision for all types of DDoS attacks. When we use matching signature method as the evaluated result of the model, the result shows that the model can detect LDAP-SNMP, MSSQL, NetBIOS, UDP-SSDP, and SYN DDoS attacks. Notice that HTM-KNN model cannot distinguish between LDAP and SNMP; UDP and SSDP with the selected features. In our experiment, True-Positives and False-Positives of UDP-Lag is 0, so the Detection rate is 0 and Precision is 0/0. Refer to attack signatures present in table I, UDP-Lag has the same attack signature as normal traffic pattern, so that the model didn't detect any UDP-Lag pattern. In order to detect UDP-Lag DDoS attack and more types of attack, we will continue research to add new features to feature set. Those new features should show changes when attacks appear.

Table IV presents the confusion matrix of model. Again, there are some false negatives in the time intervals of LDAP-SNMP, MSSQL, NetBIOS, UDP-SSDP, and SYN DDoS attacks. The false positives are due to the misses of the matching signature of observed attack patterns. As a result, the model labels no matching. Instead of assigning labels for no matching pattern, we can adapt the strategy by referring the most similar attack signatures. The most similar attack signature can be specified based on distance between SDRs of the closest attack signatures with the observed patterns. Figure 5 and Figure 6 present the classification results of the HTM-KNN model for the entire observing time. The model uses matching signature method and most similar attack signature method to classify patterns. The orange columns present the predicted class of the HTM-KNN model, and the blue lines present the Actual class of the CICDDoS 2019 testing dataset. For each time interval, when the blue line matches the top edge of the column, it means that model has a correct prediction. In matching signature method, if an observed attack pattern matches a prototype pattern (distance is zero), it will be labeled the same as the prototype pattern. The observed attack pattern will be labeled as "No Matching", if the model cannot find any matching prototype pattern. In most similar attack signature method, model will assign label of the closest prototype pattern to observed attack pattern.

The model can distinguish the attack time interval from the normal. Using the matching signature method, the model achieves higher accuracy for detecting the normal behavior time intervals. Moreover, the most similar attack signature method can achieve a higher detection rate.

TABLE III
THE PERFORMANCE OF THE HTM-KNN MODEL.

Metric	Attack types	HTM-KNN
Detection rate (%)	LDAP-SNMP	91
	MSSQL	77
	NetBIOS	87
	UDP-SSDP	80
	UDP-Lag	0
	SYN	90
Accuracy (%)	LDAP-SNMP	99
	MSSQL	99
	NetBIOS	99
	UDP-SSDP	99
	UDP-Lag	96
	SYN	99
Precision (%)	LDAP-SNMP	100
	MSSQL	100
	NetBIOS	100
	UDP-SSDP	89
	UDP-Lag	0/0
	SYN	100

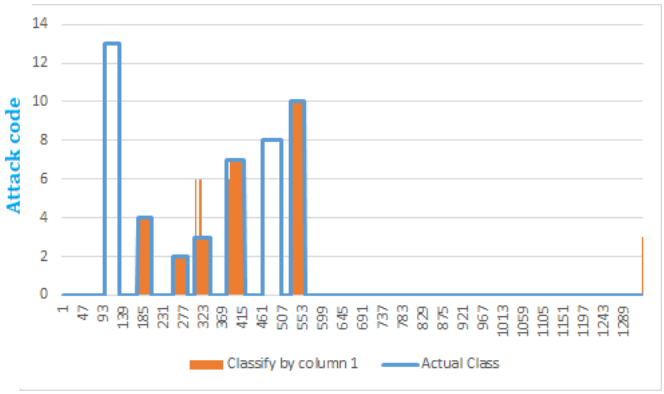

Fig. 5. Classify time intervals and assign label using matching signature method

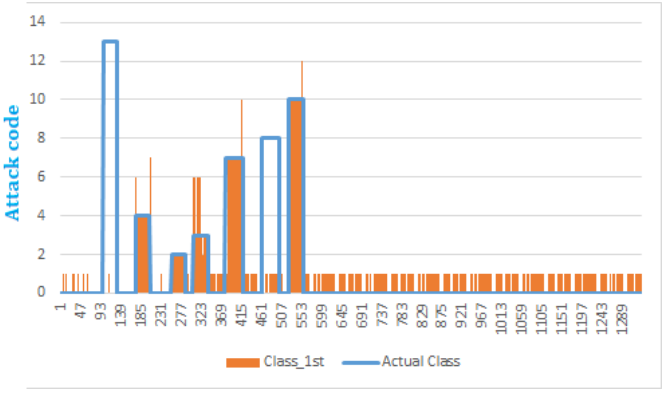

Fig. 6. Classify time intervals and assign label using most similar attack signature method

TABLE IV
CONFUSION MATRIX OF TESTING DATA SET USING HTM-KNN MODEL

Actual Class	Predicted Class											
	NTP	DNS	LDAP-SNMP	MSSQL	Net-BIOS	SSDP-UDP	UDP-Lag	Web-DDoS	SYN	TFTP	Normal	No Matching
NTP	**0**	0	0	0	0	0	0	0	0	0	0	0
DNS	0	**0**	0	0	0	0	0	0	0	0	0	0
LDAP-SNMP	0	0	**30**	0	0	0	0	0	0	0	0	3
MSSQL	0	0	0	**28**	0	4	0	0	0	0	2	2
NetBIOS	0	0	0	0	**29**	0	0	0	0	0	0	4
SSDP-UDP	0	0	0	0	0	**33**	0	0	0	0	1	7
UDP-Lag	0	0	0	0	0	0	**0**	0	0	0	27	14
Web-DDoS	0	0	0	0	0	0	0	**0**	0	0	0	0
SYN	0	0	0	0	0	0	0	0	**30**	1	0	2
TFTP	0	0	0	0	0	0	0	0	0	**0**	0	0
Normal	0	0	0	0	0	0	0	0	0	0	**761**	0

VI. CONCLUSION AND FUTURE WORKS

This paper proposes methods to classify different types of DDoS attacks using entropy-based features, Hierarchical Temporal Memory (HTM), and K-nearest neighbors algorithm. The methodology adapts the Shannon entropy as an essential indicator to detect DDoS attacks in real-time. HTM allows the model to remember all prototype patterns of different types of attacks and assign labels for input patterns by other machine learning algorithms, including KNN. The models also can implement incremental learning by updating prototype patterns without retraining the entire model. The experiment is mainly conducted by using the merged CICDDoS 2019 and MAWI dataset. The simulation results show that the proposed models have high performance classifying LDAP, SNMP, MSSQL, NetBIOS, UDP, SSDP, SYN DDoS attacks. We plan to update the model by adding more features to detect more different types of attacks based on different network traffic traces for future work. We also want to complement new features to discriminate between SSDP and UDP DDoS attacks and implement the model on physical switches to detect DDoS in real-time.

REFERENCES

[1] Y. Cui, S. Ahmad, and J. Hawkins, "The HTM Spatial Pooler - A Neocortical Algorithm for Online Sparse Distributed Coding," *Frontiers in Computational Neuroscience*, vol. 11, p. 111, Nov. 2017.

[2] Y.-K. Lai, P.-Y. Huang, H.-P. Lee, C.-L. Tsai, C.-S. Chang, M. H. Nguyen, Y.-J. Lin, T.-L. Liu, and J. H. Chen, "Real-Time DDoS Attack Detection using Sketch-based Entropy Estimation on the NetFPGA SUME Platform," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Auckland, New Zealand, Dec. 2020.

[3] S. Daneshgadeh, T. Ahmed, T. Kemmerich, and N. Baykal, "Detection of DDoS Attacks and Flash Events Using Shannon Entropy, KOAD and Mahalanobis Distance," in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, Feb. 2019.

[4] A. Koay, A. Chen, I. Welch, and W. K. G. Seah, "A new multi classifier system using entropy-based features in DDoS attack detection," in *2018 International Conference on Information Networking (ICOIN)*, Jan. 2018, pp. 162–167.

[5] X. Ma and Y. Chen, "DDoS Detection Method Based on Chaos Analysis of Network Traffic Entropy," *IEEE Communications Letters*, vol. 18, no. 1, pp. 114–117, Jan. 2014.

[6] I. Sumaiya Thaseen and C. Aswani Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, Oct. 2017.

[7] K. A. Jalil, M. H. Kamarudin, and M. N. Masrek, "Comparison of Machine Learning algorithms performance in detecting network intrusion," in *2010 International Conference on Networking and Information Technology*, Jun. 2010, pp. 221–226, iSSN: 2324-819X, 2324-8203.

[8] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, "Practical real-time intrusion detection using machine learning approaches," *Computer Communications*, vol. 34, no. 18, pp. 2227–2235, Dec. 2011.

[9] Y. Fu, F. Lou, F. Meng, Z. Tian, H. Zhang, and F. Jiang, "An Intelligent Network Attack Detection Method Based on RNN," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, Jun. 2018, pp. 483–489.

[10] F. Meng, Y. Fu, F. Lou, and Z. Chen, "An Effective Network Attack Detection Method Based on Kernel PCA and LSTM-RNN," in *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, Dec. 2017, pp. 568–572.

[11] P. Khuphiran, P. Leelaprute, P. Uthayopas, K. Ichikawa, and W. Watanakeesuntorn, "Performance Comparison of Machine Learning Models for DDoS Attacks Detection," in *2018 22nd International Computer Science and Engineering Conference (ICSEC)*. Chiang Mai, Thailand: IEEE, Nov. 2018, pp. 1–4.

[12] X. Qin, T. Xu, and C. Wang, "DDoS Attack Detection Using Flow Entropy and Clustering Technique," in *2015 11th International Conference on Computational Intelligence and Security (CIS)*, Dec. 2015.

[13] E. Balkanli, J. Alves, and A. N. Zincir-Heywood, "Supervised learning to detect DDoS attacks," in *2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*. Orlando, FL, USA: IEEE, Dec. 2014, pp. 1–8.

[14] J. Mao, W. Deng, and F. Shen, "DDoS Flooding Attack Detection Based on Joint-Entropy with Multiple Traffic Features," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, Aug. 2018.

[15] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*. CHENNAI, India: IEEE, Oct. 2019, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8888419/

[16] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking," in *ACM CoNEXT '10*, Philadelphia, PA, December 2010. [Online]. Available: http://www.fukuda-lab.org/mawilab/