Securing IoT based Maritime Transportation System through Entropy-based Dual-Stack Machine Learning Framework

Farhan Ali[1], Sohail Sarwar[5], Qaisar M. Shafi[1], Muddesar Iqbal[2], Muhammad Safyan[3], Zia Ul Qayyum[4]

***Abstract*—Internet of Things (IoTs) is envisaged to widely capture the realm of logistics and transportation services in future. The applications of ubiquitous IoTs have been extended to Maritime Transportation Systems (MTS) that spawned increasing security threats; posing serious fiscal concerns to stakeholders involved. Among these threats, Distributed Denial of Service Attack (DDoS) is ranked very high and can wreak havoc on IoT artifacts of MTS network. Timely and effective detection of such attacks is imperative for necessary mitigation. Conventional approaches exploit entropy of attributes in network traffic for detecting DDoS attacks. However, majority of these approaches are static in nature and evaluate only a few network traffic parameters, limiting the number of DDoS attack detection to a few types and intensities. In current research, a novel framework named "Dual Stack Machine Learning (S2ML)" has been proposed to calculate distinct entropy-based varying 10-Tuple (T) features from network traffic features, three window sizes and associated Rate of Exponent Separation (RES). These features have been exploited for developing an intelligent model over MTS-IoT datasets to successfully detect multiple types of DDoS attacks in MTS. S2ML is an efficient framework that overcomes the shortcomings of prevalent DDoS detection approaches, as evident from the comparison with Multi-layer Perceptron (MLP), Alternating Decision Tree (ADT) and Simple Logistic Regression (SLR) over different evaluation metrics (Confusion metrics, ROCs). The proposed S2ML technique outperforms prevalent ones with 1.5% better results compared to asserted approaches on distribution of normal/attack traffic. We look forward to enhance the model performance through dynamic windowing, measuring packet drop rates and infrastructure of Software Defined Networks (SDNs).**

***Index Terms*— Intelligent Maritime Transportation Systems (MTS), Distributed Denial of Service Attack (DDoS), Dual-Stack Machine Learning, Entropy Features**

I. Introduction

The far-ranging impact of the Internet of Things (IoTs) is expected to grow manifold where connected devices would increase up to 64 billion by 2025 from 10 billion in 2018 [1]. This mammoth increase in connected devices would be courtesy to connected artifacts over transportation media including Maritime Transportation Systems (MTS) as autonomous units. The MTS has potential of harnessing the comfort and ubiquity for effective services at optimal cost [2]. The promising applications of IoT based MTS are materializing as integral component of smart maritime infrastructure. The IoT associated MTS applications (exemplified through multimedia, navigation, autonomous controls wearables and bio-sensors etc.) collect, process and transmit sensitive/critical information over the MTS networks. Moreover, MTS contributes upwards of 500 billion dollars to US economy only [3]. Barring these vast applications and benefits, it's all about the connectivity of tiny sensors over the internet with flexibility of device control and management, remotely. This flexibility of services over the internet comes with associated network security threats and hence to the personal information of end-user-services.

One of the common threats to service provision in MTS networks is Denial of Service (DoS) attacks [4, 5]. The intention of DoS attacks is to over-engage the devices/network in such a way that actual users remain deprived of legitimate services [6]. In DoS, the network is flooded with an enormous number of service-requests that overloads the resources preventing all legitimate requests from being fulfilled. A Distributed Denial-of-Service attack (DDoS attack) is another variant of DoS attacks [7]. In DDoS, multiple sources superfluously flood the service provider for non-availability of services to intended service seekers as shown in Fig 1. The number of cyber-attacks in maritime transport increased by 400% in 2020, according to the firm named *Naval Dome*, requiring suitable counter-measures [8].

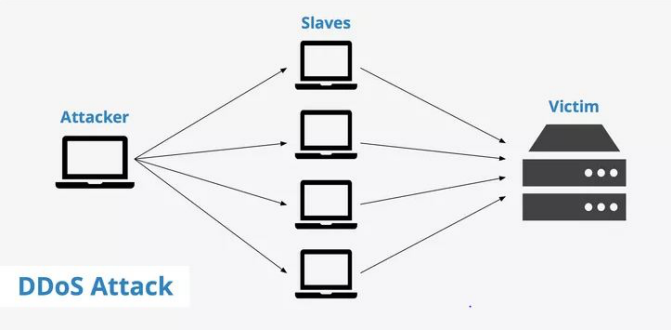

Fig 1: Flow of a DDoS Attack in MTS Network

[1] University of Engineering and Technology (CASE), Pakistan
[2]School of Engineering, Computer Science and Informatics Division London South Bank University (LSBU), London, England SE1 0AA.
[3] Department of Computer Science, GC University Lahore
[4] Department of Computer Science, Allama Iqbal Open University, Pakistan
[5] Renewable Energy Lab, Communications & Networks Engineering Department, College of Engineering, Prince Sultan University, Riyadh 11586, Saudi Arabia

DDoS attacks can be mitigated using different techniques such as presented in [9, 10]. One of the most promising approaches is the use of Entropy [8] (a measure of uncertainty or randomness in a system). Higher the amount of disorder in a system, higher will be its entropy. Therefore, entropy can be a good measure for differentiating normal traffic in an IoT network from DDoS attack traffic. This is a fundamental step in many techniques proposed for detecting anomalies in a MTS network.

A major drawback of the existing DDoS detection techniques based on Entropy is that they focus mainly on analyzing the entropy of static 4-tuple network parameters [11, 12] i.e. Source IP Address, Destination IP Address, Source Port and Destination Port for the detection of attack traffic. Due to such static and limited parameters, numerous DDoS attacks of different types and varying intensities cannot be detected. Such attacks can jeopardize the MTS services over the network. Moreover, there is no generic security framework with mutual agreement among all stakeholders for MTS. So, dynamic and generic approaches from the realm of Machine Learning (ML) were exploited in current research to cater the asserted challenges. These techniques have the ability to automatically learn and improve from experience without being explicitly programmed but fewer features have been used as elaborated in section III. Lastly, detection of DDoS attacks using a comprehensive MTS-IoT dataset still has not been addressed thoroughly, to the best of our knowledge.

Keeping above in view, a generic, comprehensive and intelligent framework named "S2ML" has been proposed, in this research to detect DDoS attacks in MTS infrastructure. In proposed framework, the entropy of different traffic-features in MTS network has been analyzed such as IP Addresses, MAC Addresses and entropy variation based features while extending approach by Abigail Koay [11]. Moreover, the effectiveness of proposed framework has been enhanced through entropy variation features calculated using the technique by Xinlei and Yonghong Chen [7] for developing 10-Tuple (10T) features, which yield significantly better results. A thorough evaluation of technique has been carried out over one of the promising datasets across the globe i.e. UNSW IoT Datasets [8] using three different window sizes (the time window in seconds for capturing traffic).

The salient contributions of current research have been listed below:

- A generic and dynamic framework named *S2ML* has been proposed to cater DDoS attacks on MTS
- Entropy based traffic 10-Tuple features used for IoTs in MTS networks
- An effective approach based on supervised learning has been implemented
- A thorough analysis of Rate of Exponent Separation (RES) based approaches and varying network traffic windows size
- A comparison of proposed S2ML with prevalent techniques namely MLP, SLR and ADT
- Effectiveness of proposed approach using different metrics with different parameters

The organization for rest of the paper is given as: section 2 reviews literature in IoT/MTS/ML applications, entropy/ML-based DDoS detection techniques for MTS networks. Section 3 gives a rationale to the discussion on how network features were calculated for detecting DDoS attacks. Section 4 furnishes the proposed approach S2ML. Results are given in section 5 that comprehensively explains the accuracy of the proposed technique in detecting DDoS attacks when applied over three MTS-IoT datasets. Section 6 concludes the work presented with conclusive review of and potential future work.

II. Literature review

IoTs ubiquitously connect network devices without constraining time and location with machine-to-machine (M2M) learning. IoT technology has a very vast number of applications that have made a huge impact in human life. Some of the pertinent IoT applications are Smart Homes, Smart Farming, Smart Cities [10], Connected Industry [11], logistics and transportation [12] etc.

A. IoT Security

IoT security deals with securing and safeguarding the devices and networks in the IoT domain. IoT in MTS security has come under a great amount of scrutiny after numerous breaches of security have taken place via a common IoT device by using it to penetrate and attack the network [8]. Detecting such attacks and implementing appropriate security measures are therefore essential for ensuring the safety of IoT networks and all the devices within them.

A major issue is the presence of hard-coded usernames and passwords in the devices. Moreover, IoT attacks don't target the IoT device itself but use it just as an entry point into the network [12]. It makes these devices very easy to access for hackers. IoT devices often have very limited computing resources which makes it impossible to implement strong security e.g. temperature and humidity sensors cannot handle advanced encryption or other effective security measures. Moreover, manufacturers don't roll out security firmware updates regularly which makes the devices vulnerable to emerging threats.

B. Security of IoT MTS Network

In [12], a review of cyber-attacks has been presented in maritime sector over the period of last 20 years with major impact on world economy. It discusses 90 publicly reported cyber-attacks on MTS networks. These cyber invasions targeted maritime sector with various motives of cyber attackers such as: abuse of data, stealing money/cargo, service disruptions, spying for information and weakening economy etc. For example 6 to 15 ports in Asia-Pacific were closed due to DDOS attacks causing estimated damage of $80 – $219 billion.

There are different modules within a MTS that can be targeted by DDOS attackers such as GPS, navigation system, Automatic Identification system, steering systems, transceivers, maritime

cargo trackers, optical recognition to manage port operations etc. The synergy, inter-operation and inter communication of all these modules is of pivotal importance for smooth functioning of IoTs based MTS networks. A timely detection and mitigation of these threats is imperative to prevent socio-economic losses.

There is no single IoT security framework in existence upon which there is a mutual agreement among all involved manufacturers for MTS infrastructure. So a generic, dynamic and adaptive framework is desired to ensure end-to-end operational MTS infrastructure.

C. Entropy and Information

Entropy is defined as the measure of energy in an object, which is unavailable to be utilized for doing some work. In general, we can say that it is a measure of randomness and uncertainty in a system [13]. The higher the entropy of an object, the more uncertain we will be about the state of that object. For a system, entropy is directly proportional to chaos in the system. While considering the Entropy of information, the amount of information in an event is inversely proportional to its certainty. This means that the more deterministic an event is, the less the information it will contain. Anukool et al [14] have used entropy as a summarization tool to detect anomalies in network traffic. They have shown that by analyzing the distribution of two network features i-e IP Address and Port, the presence of anomalies can be detected.

D. DoS, DDoS Attacks and Machine Learning

A Denial of Service (DoS) attack in a computing environment is a cyber-attack in which the attacker intends to cause a machine or a network service to become unavailable to its users by stopping the services of a host connected to the internet momentarily or indefinitely. It is achieved by overloading the systems with superficial or illegitimate requests which flood it ultimately prevent actual requests from being responded.

In a DDoS, the flood of traffic which overwhelms the victim originates from more than one source. A thorough analysis was performed by Hodo et al [16] for preventing IoT threats using Artificial Neural Network (ANN). The training was performed on a Multilayer Perceptron (MLP), a type of supervised ANN, using a simulated IoT network having only five nodes. A UDP flood DDoS attack was simulated and then the proposed technique was evaluated.

In [17], Rohan et al proposed a technique for detecting DDoS attacks in IoT network comprising of consumer products by using network packet features like packet size, inter-packet time interval, bandwidth and number of destination IPs to construct feature vectors and then classify these using five different ML techniques [4, 15, 16]. They set up an IoT network comprising of a router, some IoT consumer devices for normal traffic and some for attack traffic. A single dataset of about 490,000 packets was produced this way. The shortcoming of their approach is that it can detect only a few types of DDoS attacks. Moreover, its robustness and diversity is not proven yet as it has been tested on a very small and uniform dataset.

It is very difficult to distinguish DDoS traffic from legitimate network traffic by simple means. Hackers can bypass security measures by placing random values in the IPv4 packet's source fields. Entropy can be used to represent the randomness in network traffic effectively which can help in utilizing it to detect DDoS attacks more efficiently as compared to signature-based methods.

As a statistic metric, entropy has been used in anomaly detection by many researchers. It describes the degree of concentration and dispersal characteristics of the traffic. Generally, entropy-based detection techniques depend only on the values computed by each packet field, while the connection information or the relationship between each field is not taken into consideration.

DDoS attack detection experiments were performed using chaos theory [14] in the MIT Dataset [18, 19].

Stephen at el. [20] revealed that linking with remote sensor networks is a phone to sinkhole assaults. This Sinkhole Attack diminishes the stream of traffic, bantering the Senders and network that provided the packet to its intended destination. This attack is a complex assault that can help lead to a Denial of Service (DoS) attack by causing traffic and interrupting the routing route [20]. An interruption detection system used PRL protocol and aware of the foliage to decrease packet loss.

A Versuche to detect IoT-based attacks projected Message Queuing Telemetry Transport (MQTT) transaction-based features is presented by Moustafa et al. [18]. The authors, however, recycled features based on the TCP protocol analysis, that don't have adequate details about the MQTT protocol parameters. Our proposed MQTT features, on the other hand, are based on MQTT header and payload meta-data which can detect and distinguish these attacks effectively. Moreover, Mustafa et al. [21] has come up with the main drawback is that the Quality of their MQTT attack detection scheme was not presented. The main reason behind this was that no specific MQTT attack datasets were available to check the detection techniques. In this work, the first pose various vulnerabilities in MQTT and then create several attack scenarios to generate real DoS attack traffic. The author also tests the capability of the proposed IoT detection system for the attack.

The IoT signals are obtained using sensors that are connected to the patient. The system's effectiveness greatly depends on the sensor network's performance. Wu et al. [22] suggested an integrated network of sensors to track health care. This network can communicate through the gateway system called a LoRa network between sensors from different subjects. This gateway makes use of Bluetooth to communicate between sensors. The data obtained from the nodes will be stored on a cloud server and processed there. This system's main purpose is to boost network classification accuracy by linking all of the sensors in sequence.

Pirbhulal et al. [23] suggested a heartbeat sequence-based safety method. They created binary heartbeat sequences using interpulse interval values. They created the 128-bit binary sequence of sequence in 8 s from MIT-BIH Arrhythmia database. So they reduced the time taken for random binary sequences to be generated. The biggest concern with the use of

heartbeat for safety over time is its lack of precision.

Kumari and Anjali [24] suggested a double encryption scheme to secure node and base station communication. They use simple mathematical functions to encrypt the data, rather than complex mathematical formulas. For this reason, the system uses complex mathematical functions to consume less time compared to other techniques. It is often considered a drawback because replicating the key is simple for attackers [25-28].

In summary, every IoT based infrastructure (personal, smart cities, health care, logistics, transportation etc.) is vulnerable to DoS/DDoS attacks that can potentially cause irrecoverable economic losses. The issues that have been highlighted in different techniques need to be addressed such as dispersal characteristics of network traffic, signature based methods, entropy based approaches and protocol parameters based approaches. The timely prevention, detection and mitigation of these attacks require novelty of catering attacks on IoT based MTS with enhanced\cohesive feature-sets, maximum coverage of scenarios through updated datasets and dynamic models.

III. Proposed Framework

All the steps involved in the Rate of Separation based algorithm have been discussed in the following. Moreover, the procedure for sorting out DDoS attack traffic from MTS network traffic has been explained.

A. Overview of the Algorithm

The proposed algorithm comprises of the following three steps:

Step 1: Calculate the Entropy of source IPs and Destination IPs.

Step 2: Calculate the Rate of Exponent Separation (RES).

Step 3: Define the range for RES variable to detect the DDoS attack.

Each step is explained below

Step 1:

In this step, Entropy is calculated for a given probability distribution $P = \{P_1, P2, .P_N\}$ where $0 \leq P_i \leq 1$ using the following equation as given in [4]:

$$H_q = \frac{1-\sum_{i=1}^{N} P_i^q}{q-1} \quad (3.1)$$

where,

H_q = *Tsallis Entropy.*

q = *Entropic Parameter (any positive number)*

N = *Number of Packets in the Dataset.*

P_i = *Probability of the* i_{th} *event.*

The value of q causes a change in the relative contribution of the given event to the whole event. The value of entropy ranges from 0 to H_q^{max} , which represents maximum dispersion and maximum concentration. The maximum value of entropy H_q^{max} [3] is defined as:

$$H_q^{max} = \frac{1-N^{1-q}}{q-1} \quad (3.2)$$

Step 2:

In this step, first of all, the entropy of the observed network traffic is normalized with respect of maximum entropy. The normalized entropy $H_{norm\ is}$ given by:

$$H^{norm} = \frac{H_q}{H_q^{max}} \quad (3.3)$$

These normalized entropy sequences are calculated for both source IPs and destination IPs. In Fig 2, the entropy for the source IPs and destination IPs in the network traffic as observed in [4] has been illustrated.

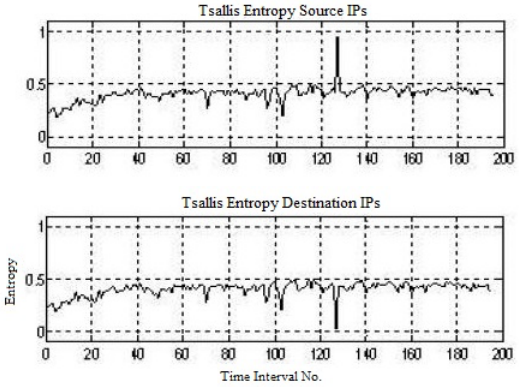

Fig 2: Tsallis Entropies for Source and Destination IPs

The Rate of Exponent Separation λ_k is then calculated using:

$$\lambda_k = \frac{1}{t_k} \ln \frac{H_s(k)}{H_d(k)} \quad (3.4)$$

where,

H_s = H_{norm} of source IPs

H_d = H_{norm} of destination IP

t_k = *Interval number starting from 1,2,3……*

In Fig 3, the RES for the entropies of source IPs and destination IPs shown in Fig 2 can be seen.

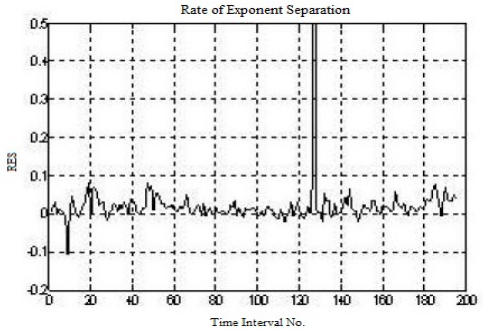

Fig 3: Rate of Exponent Separation

Step 3:

In this final step, the attack is detected by analyzing the values of λ_k. When $\lambda_k > 0$, the entropy values for Source IPs are greater than those of Destination IPs. This means that a dispersal trend exists among the Source IPs in the Dataset. The

degree by which $\lambda_k > 0$ determines whether the traffic is normal or not. When $\lambda_k < 0$, the entropy values for destination IPs are larger than source IPs. It means there's a dispersal trend among the destination IPs. A classification threshold λ^* is determined to separate out normal traffic from attack traffic. Firstly, Rate of change of λ_k is determined by taking its derivative. Then, the threshold λ^* is set based on observations on network traffic.

B. Implementation of Algorithm

The implementation level details for feature extraction from network traffic have been discussed in this section.

Calculation methods for entropy and subsequently the rate of exponent separation based features are explained. Then the Machine Learning based classification phase of the technique is elaborated i.e. Proposed S2ML algorithm is explained.

1) Extraction of Network Features from Packets

The first step of the analysis is to extract all the required information from the provided pcap files of the sub-dataset. Table 1 contains information about the features that have been extracted.

TABLE1: TRAFFIC FEATURES FOR EACH PACKET EXTRACTED FROM THE RAW TRAFFIC DATA.

No.	*Feature*	*Description*
1.	*Source IP Address*	*Source IP Address of the packet.*
2.	*Destination IP Address*	*Destination IP Address of the packet.*
3.	*Source MAC Address*	*Hardware address of the previous network router/host the packet is coming from.*
4.	*Destination MAC Address*	*The hardware address of the next-hop router/host to which the packet is headed.*
5.	*Source Port Address*	*Source Port Number of the packet.*
6.	*Destination Port Address*	*Destination Port Number of the packet.*
7.	*Protocol*	*Type of protocol being used e.g. HTTP, TCP etc.*

2) Entropy Calculation

The next step is to specify a window size W, e.g. 30s, for the network traffic and calculate the entropy of each traffic feature as presented in Table 1. This window is slid along until the end of the dataset. Entropy is calculated using Shannon's Entropy formula [16] which is:

$$H = -\sum_{i=1}^{n} P_i \log P_i \qquad (4.1)$$

where,

H = Shannon's Entropy.

n = Number of Packets in the window W.

P_i = Probability of the Network Feature within the window e.g a particular IP address.

Shannon's Entropy formula has been used instead of the Tsallis Entropy formula because it provides greater magnification for entropy variations. A comparison between these entropy formula outcomes for the same set of data can be observed in Figs 4 and 5 respectively. It can be concluded that Shannon's Entropy provides much more clear variations.

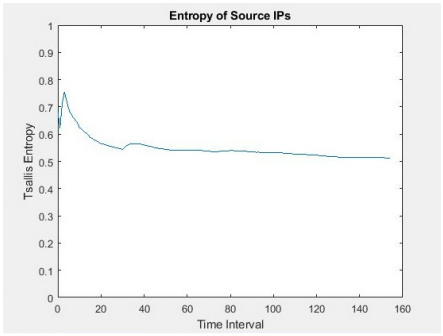

Fig 4: Tsallis Entropy for Source IP Addresses with W=30s.

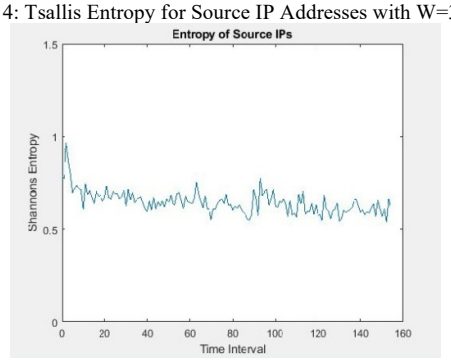

Fig 5: Shannon's Entropy for Source IP Addresses with W=30s.

After calculation of Shannon's Entropy, it is normalized the same way as in section 3.1 using H_q^{max}, which is calculated using eq 3.2.

3) Calculation of Entropy Variation Features

The next step is to calculate the entropy variation features. An entropy variation feature is found out by calculating the RES in the very same way as for Source and Destination IPs in section 3.2. Table 2 shows the entropy variation features used:

TABLE 2: ENTROPY VARIATION FEATURES CALCULATED USING RES.

No.	*Feature*	*Description*
1.	*Separation IP*	*RES between source and destination IPs.*
2.	*Separation MAC*	*RES between source and destination MACs.*
3.	*Source PORT*	*RES between source and destination PORTs.*

4) Detection of an Attack using S2ML

The last phase of the proposed framework S2ML targets to detect an attack based on model parameters as elaborated in prior sections. S2ML using a combination of three ML classification algorithms.

The three algorithms are:

- Multilayer Perceptron (MLP)
- Alternating Decision Tree (ADT)
- Simple Logistic Regression (SLR)

A simple ensemble-based approach, which is called stacking, is applied. The detection is divided into 2 phases. In phase 1, ADT and SLR jointly classify the data in the dataset using the majority voting scheme. Each contributes a vote for the data to be either attack or benign as shown in Fig 6.

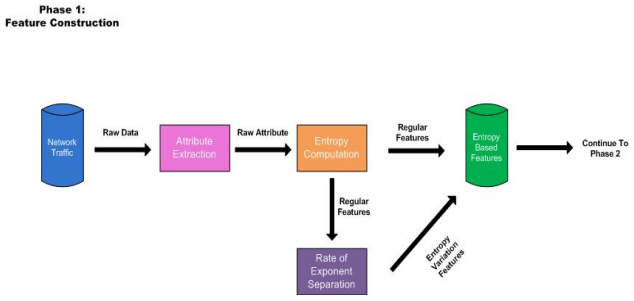

Fig 6: Phase 1 of proposed framework.

In phase 2, the classified data is again classified by the MLP algorithm. The parameters for MLP i.e. input layer, neurons on input layer, the hidden layers and output layers with respective neurons were adjusted through experimentation. This, as will be proven by the results of the experiments in the next chapter, yields better results as compared to classification using a single technique. Detailed analysis and comparison of results using only the traditional 4 tuple entropy features (Source and Destination IP address, Source and Destination Ports) and the 10 tuple features have been mentioned at the beginning of section 3. Fig 7 shows the second phase and the flow of the algorithm.

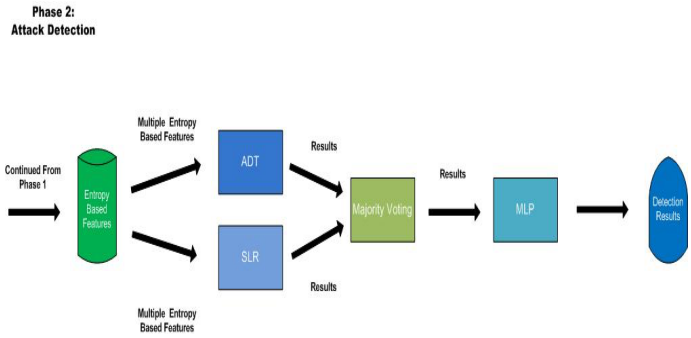

Fig 7: Phase 2 of the proposed framework.

IV. Results and Evaluation

In this section, a detailed view of two datasets is provided that have been used for the experiments, along with the experimental setup. The experiment results and observations are explained and in the end, the results are presented and analyzed.

A. The *MTS-IoT* Dataset

The dataset which was used for testing and evaluation of the proposed DDoS attack was obtained from [5]. The researchers at the University of New South Wales set up an IoT network comprising of more than 28 devices including smart lights, cameras, motion sensors, appliances and real time monitoring devices. They synthesized different scenarios of network traffic for 6 months (free for anyone to use) and collected traces of it and performed different research techniques on it. The datasets obtained were ".pcap" files i.e. Wireshark dumps of network traffic.

Entropy and RES based entropy variation features were calculated for each window size and written back into the CSV files in new columns using MATLAB. The CSV files containing the entropy and RES based features were loaded into WEKA and then different ML algorithms were applied on it and their performance was evaluated. These were analyzed to see and observe the different DDoS attacks taking place.

Following 2 sub-datasets, which will be referred to as datasets, later on, have been used for analysis and performance comparison:

- 18-06-01 (Dataset 1)
- 18-06-02 (Dataset 2)

B. *Dataset 1 (18-06-01)*

This dataset contains a total of 450,000 packets. Two types of DDoS attacks have been simulated in this dataset. These are explained below:

1) ARP Spoofing Attack

It is a type of attack in which the attacker transmits modified ARP (Address Resolution Protocol) messages over the LAN. The target of such an attack is to associate the MAC address of the attacker with the IP address of a legitimate user of the network. This causes the attacker to receive data intended for the legitimate user. Wire-shark dump of this dataset shows that the device with IP address 192.168.1.205 is continuously broadcasting "who has" requests with incremental IP addresses. Such traffic can cause congestion in the network also.

2) TCP SYN Flood

TCP syn flood attack makes use of the TCP three-way handshake to overwhelm a host and make it unresponsive. In a three-way handshake, firstly the client requests a connection by sending a SYN message to the server. The servers acknowledge this message by sending an SYN-ACK back to the client. The client then responds to the server by sending an ACK message to the server and a connection is established for further communication. In a TCP SYN Flood attack, the attacker sends SYN messages to several ports of the target server often using fake IP addresses. The server responds to each message with an SYN-ACK. The attacker does not respond to these SYN-ACK

messages. During this time the server cannot close the connection with an RST packet and the connection remains open. SYN packets keep on arriving and this leaves a very large number of connections half-open. Legitimate users don't get serviced as a result and the server may also crash.

C. *Dataset 2 (18-06-02)*

This dataset contains a total of 450,000 packets. One type of DDoS attack has been simulated in this dataset. It is explained below:

1) *SSDP Flood*

The Simple Service Discovery Protocol (SSDP) flood attack is a DDoS that makes use of the Universal Plug n Play (UPnP) protocol to send a very large amount of traffic to a target host in the network to exhaust its resources and ultimately render it.

D. *Traffic Distribution, Entropy and RES*

In this section, the calculations for the entropy of selected network features depicted in Table 4.2 and RES for the entropy variation features in Table 4.3 are calculated for each of both datasets. Three different window sizes for W were used i-e 30s, 60s and 90s. A graph for each calculated feature was also plotted.

E. *Calculations for Datasets*

In this section, the calculations for the entropy of selected network features depicted in Table 4.2 and RES for the entropy variation features in Table 4.3 are calculated for each of both datasets. Three different window sizes for W were used i-e 30s, 60s and 90s. A graph for each calculated feature was also plotted.

1) *Calculations for Datasets*

A CSV file containing the raw network features for dataset1 has been used for calculating Entropy and RES for each of the three window sizes:

a. *Traffic Distributions for Window Size W=30s, 60s, 90s*

The first thing that needs to be looked at is the traffic distribution for the size of this window. This was done by setting the size of the window to 30 and start from the time zero seconds till 30 seconds and count the number of packets in this duration. Then count from 31 seconds to 60 and so on till the end of the data. The distribution of packets for this window size can be seen in Fig 12.

Similarly, traffic distribution patterns were recorded for Dataset 2 for all three windows. It can be seen that starting from time interval number 25 to 50, a large increase in the number of packets can be seen. Similarly, two peaks of packets can be seen at the very start and around 80. These are very much the probable intervals during which the simulated attacks have taken place. The next step is the calculations of entropy values for the 7 network parameters. Entropy value is calculated for each parameter in each of the time intervals and stored in the CSV files. Fig 8 shows the plots of the entropy values of IP Addresses, MACs, Ports and Protocol respectively. The plots of entropy of MAC addresses clearly show that there is a rise in the entropy value at the same time intervals where the traffic had increased.

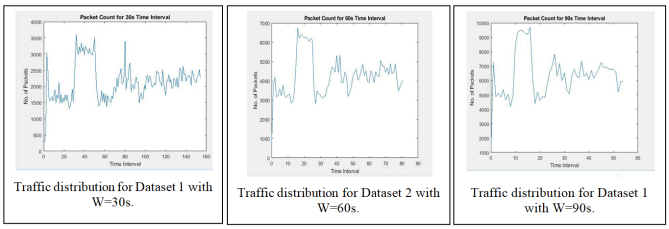

Fig 8: Traffic Distribution with Different Window Sizes

b. *Entropy for Variation based Features*

The next step is to calculate the entropy variation based features. This is done using the method explained in section 3.2. The RES values for these features are calculated and copied into the respective columns in the CSV files. The plots of RES values for IP Address, MAC Address and Ports are shown in Fig 9.

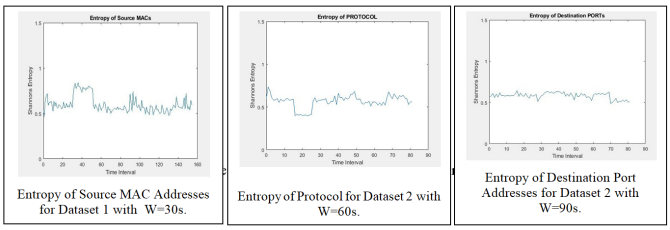

Fig 9: Entropy with Different Window Sizes

c. *Rate of Exponents Separation (RES)*

The RES value plots for IP Addresses and Ports don't give much information but looking at the RES plot for MAC Addresses, we can some variations in the same time intervals where the traffic had increased. Thus, it can be concluded that the attack traffic in these time intervals of high traffic is the attack traffic. Another column is then added in the 'csv' file containing the entropy and RES values which classifies the traffic in that interval to be either benign or attack. The attack traffic is given a value of 1 and benign is given 0. A snapshot of this file is shown in Fig 10.

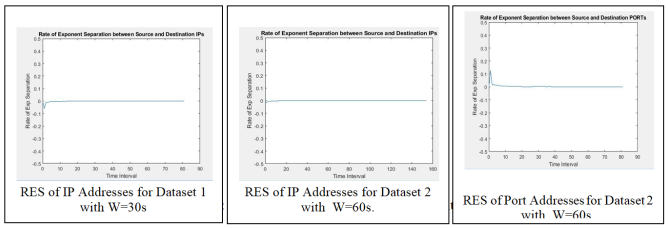

Fig 10: RES with Different Window Sizes

F. ***Experiments and Results***

This elaborates on the experiments carried out to analyze the performance of the proposed ML classification technique S^2ML against the three selected ML classification techniques. Three experiments were carried out. Evaluation parameters are explained followed by a discussion on the outcome of findings in this research.

1) Evaluation Metrics

ML techniques are evaluated in terms of some specific parameters that have been derived from Confusion metrics [16]. A brief view of interpreting the parameters is given below:

i. True Positive (TP): It is the amount of attack traffic correctly detected as an attack.

ii. False Positive (FP): It is the amount of normal traffic incorrectly detected as attack traffic.

iii. False Negative (FN): It is the amount of attack traffic incorrectly detected as normal traffic.

iv. Precision: Precision is the ratio of TP to total actual attack traffic.

v. Recall: It is the ratio of correctly detected attack traffic to the total of actual traffic.

$$Recall = \frac{TP}{TP+FN} \quad (5.2)$$

vi. F1-measure: It is the weighted average or harmonic mean of precision and recall.

$$F1-measure = \frac{2*Precision}{Precision+Recall} \quad (5.3)$$

vii. ROC area: It is the area under the Receiver Operating Characteristic Curve. The larger the area under the curve, the more useful the test is.

Dataset 1 was segregated into a training set and validation set with a ratio of 70-30. Dataset 2 was used for testing the trained algorithms. The CSV files, obtained in section 4.2, were first converted to the ".arff" format using the file viewer in WEKA, which is its native file format. Then each of the three window sizes was evaluated and results were recorded. Table 3 shows a comparison of the *F1-measure* obtained for these 10T features datasets.

TABLE 3: F1-MEASURE COMPARISON USING 10T FEATURES DATASETS.

Window Size	F1-measure			
	MLP	SLR	ADT	S²ML
30s	0.929	0.908	0.881	0.920
60s	0.954	0.951	0.924	0.963
90s	0.764	0.693	0.717	0.717

Similarly, results for ROC area are shown in Table 4

TABLE 4: ROC-AREA COMPARISON FOR AN EXPERIMENT USING 10T FEATURES DATASETS.

Window Size	ROC Area			
	MLP	SLR	ADT	S²ML
30s	0.962	0.907	0.889	0.874
60s	0.962	0.964	0.962	0.939
90s	0.760	0.933	0.689	0.689

The higher ROC values for W=60 suggests that this is the optimum window size. This experiment was then repeated for the 4T features datasets. Tables 5 and 6 show the comparison of F1-measure and ROC Area respectively.

TABLE IV5 : F1-MEASURE COMPARISON FOR EXPERIMENT 1 USING 4T FEATURES DATASETS.

Window Size	F1-measure			
	MLP	SLR	ADT	S²ML
30s	0.912	0.870	0.877	0.884
60s	0.914	0.900	0.850	0.865
90s	0.808	0.640	0.625	0.625

TABLE 6: ROC AREA COMPARISON FOR EXPERIMENT 1 USING 4T FEATURES DATASETS.

Window Size	ROC Area			
	MLP	SLR	ADT	S²ML
30s	0.938	0.908	0.825	0.812
60s	0.922	0.913	0.882	0.755
90s	0.817	0.780	0.698	0.614

To get a better look at the data shown in the tables above, graphs are plotted in Fig 11, 12, 13 to show the 10T and 4T results for F1-measure simultaneously.

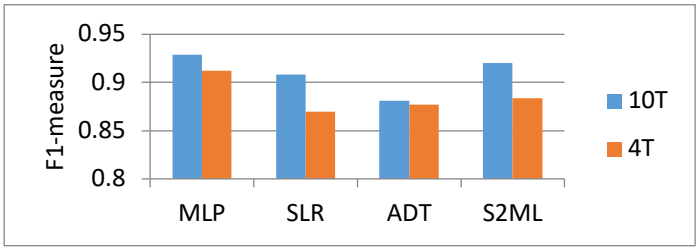

Fig11: A comparison of 10T and 4T feature results with W=30s

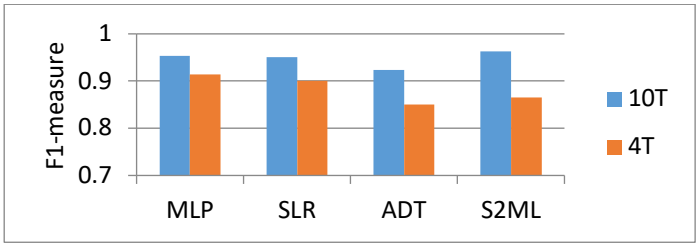

Fig 12: A comparison of 10T and 4T feature results with W=60s

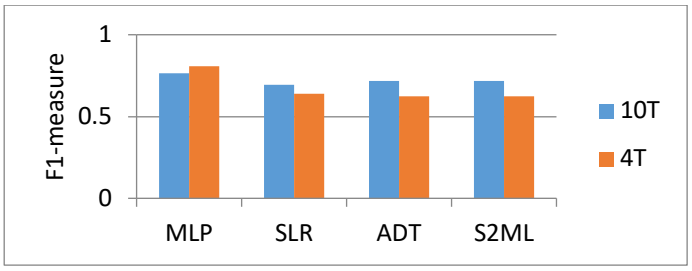

Fig 13: A comparison of 10T and 4T feature results with W=90s

From these graphs, we can conclude that 10T features give much better results for all ML techniques for all 3 window sizes. Figs 14 and 15 show a comparison of F1-measure and ROC area respectively for 10T features. These two graphs clearly show that w=60s gives the best results most of the time. It can be

concluded that the proposed S^{2}ML technique performs better most of the time compared to the other three techniques as evident from Figs 14 and 15.

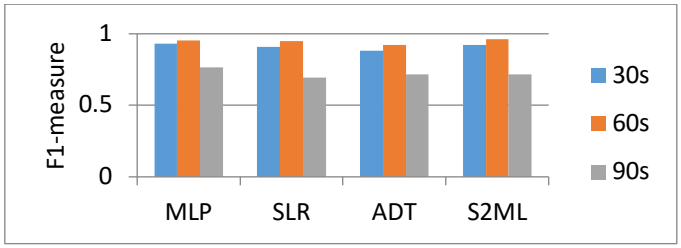

Fig 14: A comparison of F1-measure for experiment 1.

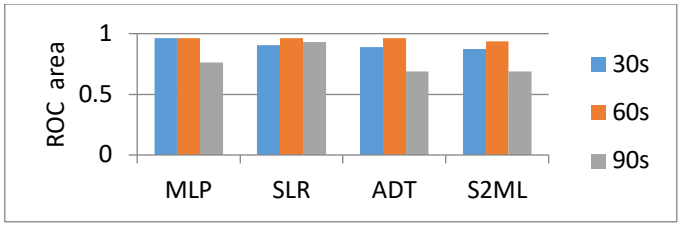

Fig 15: A comparison of ROC area for experiment 1.

All entropy-based features were calculated for all datasets and then entropy variation based features were calculated using RES for three window sizes of the 30s, 60s and 90s. All these features were saved in a CSV file. Graphs of these features were plotted to analyze them against the traffic distributions and attack traffic time intervals were determined. These intervals were marked using a class variable. Normal traffic was given a class value of 0 and attack traffic was given 1.

Results showed that the proposed S2ML technique performs on an average 1.5% better than MLP, SLR and ADT in terms of F1-measure. Moreover, the window size of the 60s was found to give the best results in this case.

Results show that the proposed S2ML framework performs on an average 1.5% better than MLP, SLR and ADT in terms of F1-measure. Moreover, the window size of the 60s was found to give the best results in this case.

V. Conclusion and Future Work

Downtime or disruption of services may not be afforded by anyone linked with MTS artifacts. Therefore, it is essential that efficient measures to detect and mitigate threats for security and performance of the MTS networks be put into place. An earnest effort has been made in this research to design a framework that effectively detects different types of DDoS attacks. The major focus has been to design and evaluate a robust DDoS detection technique that is effective regardless of the intensity and type of attack. To quantify the anomalies in MTS network traffic, entropy was selected as a basic parameter to build classification features out of raw network traffic data. In order to increase the strength of the classification technique, more features were extracted from the network packets than what has been used generally i.e. 7T features instead of 4T features. Shannon's Entropy of these features was then calculated to visualize the difference between attack and normal traffic. Three more features were added on to the 7T features by calculating the entropy separation features using the RES method. Attack traffic was then effectively sorted out from normal traffic. Comprehensive MTS-IoT datasets of the UNSW research group were used to develop 10T feature-based classified traffic data.

To evaluate the effectiveness of the proposed S2ML framework, experiments were conducted on three different datasets. Different tests using MLP, SLR, ADT and S2ML techniques were conducted for different window sizes and types of attacks. In almost all scenarios the proposed approach was found to yield better results. Moreover, it was seen that the proposed 10T features yielded significantly better results compared to the more common 4T features.

In the future, there exists a great potential in analyzing the effect of window size on the performance of the detection technique. A mechanism can be developed to adjust window size dynamically based on the density of network traffic under observation. Furthermore, there is significant headway to increase the number of features that can be extracted from a network packet while considering the packet drop rate. This will only increase the detection power of the algorithm. As network implementations are greatly heading towards Software-Defined Networking (SDN) based infrastructure, a study can be carried out to efficiently implement the proposed technique in such an environment.

Acknowledgement

We would like to thank Taif University Researchers for supporting our project number (TURSP-2020/228), Taif University, Taif, Saudi Arabia.

References

[1] Zaigham Mahmood, Connected Vehicles in the Internet of Things Concepts, Technologies and Frameworks for the IoV, Springer International Publishing, 978-3-030-36166-2, 2020

[2] Plaza-Hernández M., Gil-González, Prieto-Tejedor J., Corchado-Rodríguez, Integration of IoT Technologies in the Maritime Industry, Distributed Computing and Artificial Intelligence, 17th International Conference. vol 1242. Springer, 2021

[3] https://www.maritime.dot.gov/outreach/maritime-transportation-system-mts/maritime-transportation-system-mts, Accessed on 10 Aug 2021.

[4] Alia Mohammed Alrehan; Fahd Abdulsalam Alhaidari Machine Learning Techniques to Detect DDoS Attacks on VANET System: A Survey, IEEE International Conference on Computer Applications & Information Security (ICCAIS), 2019

[5] J. Gao, Y. Xiao, S. Rao, Security tests and attack experimentations of protoGENI. Int J Secur Network, 10(3): 151–169, 2015

[6] A. Koay, A. Chen, I. Welch and W. Seah, A New Multi Classifier System using Entropy-based Features in DDoS Attack Detection, International Conference on Information Networking (ICOIN), 2018.

[7] X. Ma and Y. Chen, DDoS Detection Method Based on Chaos Analysis of Network Traffic Entropy, IEEE Communications Letters, Vol 18(1), pp 163-178, 2014.

[8] https://www.stormshield.com/news/cybermaretique-a-short-history-of-cyberattacks-against-ports/: viewed on 25th Feb 2022.
[9] https://ahrecs.com/resources/ethics-research-important-resources-david-b-resnik-2015. : viewed on 16 Apr 2021.
[10] M. Safyan, Z. Qayyum, S. Sarwar, R. G. Castro, M. Ahmed, Ontology-Driven Semantic Unified Modelling for Concurrent Activity Recognition (OSCAR). Multim. Tools Appl, Vol 78(2), pp 2073-2104, 2019
[11] https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT: viewed on: 16 April 2021.
[12] M. Schwarz, M. Marx, H. Federrath, A Structured Analysis of Information Security Incidents in the Maritime Sector, https://arxiv.org/pdf/2112.06545.pdf ,13 Dec 2021
[13] https://internetofthingsagenda.techtarget.com/definition/IoT-security-Internet-of-Things-security: viewed on 16 January 2022.
[14] O. Irshad; M. Khan, R. Iqbal, S. Bashir; A. K. Bashir. Performance Optimization of IoT Based Biological Systems Using Deep Learning. Computer Communication, Elsevier, 2020.
[15] A. Lakhina, M. Crovella, C. Diot, Mining Anomalies Using Traffic Feature Distributions, ACM SIGCOM, Vol 35(4), pp 217–228, 2005.
[16] S. Sarwar, Z. Qayyum, O. A. Malik, A Hybrid Intelligent System to Improve Predictive Accuracy For Cache Prefetching. Expert Syst. Appl. 39(2), pp 1626-1636, 2012
[17] Rohan Doshi, N. Apthorpe and N. Feamster, Machine Learning DDoS Detection for Consumer Internet of Things, IEEE Symposium on Security and Privacy Workshops, 2018.
[18] A. Chonka, J. Singh, and W. Zhou, Chaos Theory Based Detection Against Network Mimicking DDOS Attacks, IEEE Communication Letter, Vol 13(9), pp 717–719, 2009.
[19] J. McHugh, The 1998 Lincoln Laboratory IDS Evaluation, Recent Advances in Intrusion Deteciton, pp 145–161, 2000.
[20] R. Stephen and L. Arockiam, ''Intrusion Detection System to Detect Sinkhole Attack on RPl Protocol in Internet Of Things,'' Int. J. Elect. Electron. Comput. Sci. Eng., Vol 4(4), pp 16–20, 2017.
[21] N. Moustafa, B. Turnbull, & K. Choo, An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things. IEEE Internet of Things Journal , Vol 6 (3), pp 4815–4830, 2019
[22] F. Wu, T. Wu, M. Yuce, Design and Implementation of a Wearable Sensor Network System for IoT-Connected Safety and Health Applications, IEEE 5th World Forum on Internet of Things (WF-IoT), pp. 87-90, 2020
[23] S. Pirbhulal, H. Zhang, W. Wu, Heartbeats based Biometric Random Binary Sequences Generation to Secure Wireless Body Sensor Networks, IEEE Trans. Biomed. Eng., Vol 65 (12), pp 2751-2759, 2018
[24] P. Kumari, T. Anjali, Securing a body sensor network 2017 9th International Conference on Communication Systems and Networks (COMSNETS), IEEE, pp 514-519, 2017
[25] N. Tsafack; S. Sankar; B. Abd-El-Atty; J. Kengne; K.C. Jithin; A. Belazi; I. Mehmood; A. K. Bashir; O.Y Song; A. A. EL-Lati. A new chaotic map with dynamic analysis and encryption application in Internet of Health Things. Vol 8(1), pp. 13731-13745, IEEE Access, 2021.
[26] A. Nawaz; J. P. Queralta; J. Guan; M. Awais; T. N. Gia; A. K. Bashir; H. Kan; T. Westerlund. Edge Computing to Secure IoT Data Ownership and Trade with the Ethereum Blockchain, Vol 20 (14), pp. 3965-3981, Sensors, MDPI, 2020
[27] Grzywaczewski, A., and Iqbal, R.:"Task-Specific Information Retrieval Systems for Software Engineers", Journal of Computer and System Sciences, Elsevier, Volume 78, Issue 4, pp., 1204-1218, 2012
[28] Iqbal, R., Doctor, F., More, B., Mahmud, S., Yousuf, U: "Big Data analytics and Computational Intelligence for Cyber–Physical Systems: Recent trends and state of the art applications", Future Generation Computer Systems, Elsevier, Volume 105, pp. 766-778, 2020


Farhan Ali is Research Assistant at CASE, University of Engineering and Technology Taxila. His research interests include IoT, AI for Vehicular Technologies and integration of AI with IoT applications.


Dr Sohail Sarwar received the PhD degree in Computer Science from University of Gujrat, Pakistan and did research at Universidad Technical de Madrid, Spain. His research interests include machine learning techniques, vehicular technologies, semantic technologies and knowledge engineering techniques in different applications. Currently, he is working in research position with London South Bank University, England.


Dr Qaisar Shafi received his Ph.D. & M.S. degree in information security from the National University of Sciences and Technology. He is working as Assistant Professor in SS CASE IT. He has more than ten years of experience in both academics and research. He is the lead member of Intelligent Systems Group(ISG) at SS CASE IT.


Dr Muddesar Iqbal is a Senior Lecture in Mobile Computing at London South Bank University, England. He did his PhD from Kingston University UK. His research interests include Internet of Senses Digital Twin, V2X technologies, Disaster Management and IoT. He is a visiting Research fellow at School of Computer Science and Electronic Engineering, University of Essex, UK.


Muhammad Safyan is Assistant Professor in Government College University (GCU) Lahore. He received PhD degree from National University of Sciences and Technology in 2018. His area of interest is ontology alignment, e-learning and semantic activity recognition.


Prof Zia Ul Qayyum is currently a Professor at University of Gujrat Pakistan. He received his Ph.D. degree in Computer Science from Leeds University UK in 2005. His research interests include Artificial Intelligence, Knowledge Engineering, Vehicular technologies, Data mining, Semantic web and e-learning. He is Vice Chancellor of Allama Iqbal Open University.