Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Anomaly detection framework to prevent DDoS attack in fog empowered IoT networks

Deepak Kumar Sharma [a],[*], Tarun Dhankhar [a], Gaurav Agrawal [a], Satish Kumar Singh [a], Deepak Gupta [b], Jamel Nebhen [c], Imran Razzak [d]

[a] *Department of Information Technology, Netaji Subhas University of Technology, Delhi, India*
[b] *Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology, Delhi, India*
[c] *Prince Sattam bin Abdulaziz University, College of Computer Engineering and Sciences, Saudi Arabia*
[d] *School of Information Technology, Deakin University, Geelong, Australia*

ARTICLE INFO

Keywords:
Anomaly detection
Continuous ranked probability score
DARPA-99
DDoS attack
Dimensionality reduction
Fog computing
ICMP attack
Internet of things
Kernel density estimation
SYN attack

ABSTRACT

Internet of things or in short IoT is a network of interconnected entities such as computing devices, mechanical machines, digital gadgets etc. Cloud based IoT infrastructures are susceptible to Distributed Denial of Service (DDoS) attacks. A DDoS attack may render the server useless for a long period of time causing the services to crash due to extensive load. In this project we will try to introduce the concept of fog computing and try to explain its importance in a 3-tier architecture. We have proposed an anomaly detection architecture for IoT networks where the detection actually happens on the fog layer. The algorithm is based on the CRPS metric which is a single variable algorithm which is the case in most statistical algorithms. Therefore, we have proposed a way to use multiple variables and shown why it is required in a heterogeneous network like IoT. For detection purposes(testing data) we have used Week 5 Day 1 data of DARPA 99 as it contains a TCP SYN attack initiated once for a duration of 6 min 51 s and for ICMP Week 4 Day 1 data of DARPA 99 is used it has 2 attacks for 1s each. The algorithm is able to identify these attacks correctly.

1. Introduction

Internet of things or in short IoT is a network of interconnected entities such as computing devices, mechanical machines, digital gadgets etc. These entities work with each other by either collecting or producing some data or by processing some data in order to provide a smart adaptable environment for the users. In smart home environments, all the home appliances communicate with each other to provide better living experience for e.g. a thermostat can inform the air conditioner to regulate the room's temperature. There are many such use cases of IoT networks such as smart traffic systems, smart security networks etc. It is estimated that there will be more than 21 billion IoT devices by 2025. We will be using apps to control the LEDs, door locks and other home appliances in our houses. Our cars will be able to communicate with each other to avoid road accidents. Cities will have a huge network of interconnected security cameras, alert systems etc.

But with this level of connectivity comes a great risk of security. Devices connected to an IoT network are very prone to cyberattacks. Therefore, there is a need to develop security mechanisms, attack detection systems, attack mitigation systems, secure IoT devices and connections etc. in order to make next generation IoT networks more secure and reliable. Many researchers have proposed such attack detection and mitigation techniques working at different layers of the network architecture and preventing the systems from various cyber attacks. A cyberattack over a network creates anomalous behaviour in the network traffic which means the flow of packets is different than usual and this fact can be exploited to detect attacks using various anomaly detection algorithms. Many anomaly detection techniques based on artificial intelligence, neural networks, statistics etc. have been used and these techniques have shown promising results against many attacks such as phishing attacks, DDoS attacks, sinkhole attacks etc. In this work we will propose a similar but efficient anomaly detection technique to prevent DDoS (Distributed Denial of Service) attacks in fog-empowered IoT networks.

In the past, many researchers have proposed various DDoS detection and mitigation techniques [1–4]. Machine learning algorithms and Artificial neural networks are very popular and useful choices for detecting such protocol based attacks in wireless networks. In [5], Wang et al. used a semi supervised clustering algorithm to detect DDoS attacks. In

[*] Corresponding author.
E-mail addresses: dk.sharma1982@yahoo.com (D.K. Sharma), tdhankhar@gmail.com (T. Dhankhar), gauravagrawal.2626@gmail.com (G. Agrawal), satishsingh23@gmail.com (S.K. Singh), deepakgupta@mait.ac.in (D. Gupta), j.nebhen@psau.edu.sa (J. Nebhen), imran.razzak@deakin.edu.au (I. Razzak).

https://doi.org/10.1016/j.adhoc.2021.102603
Received 15 April 2021; Received in revised form 21 June 2021; Accepted 6 July 2021
Available online 10 July 2021

this method, 3 features are selected by analysing the characteristics of the attack to form a detection feature vector. Then a multi-feature based constrained k-means algorithm is used which improves the convergence speed and the accuracy of the detection mechanism. In [6], Sumathi et al. proposed a DDoS detection mechanism using neural networks. Their model evaluates the network traffic using a deep learning classifier based on cost minimization strategy for publicly available datasets. They used detection accuracy, cost per sample, packet loss,average delay, packet delivery ratio, overhead and throughput as evaluation metrics for the performance analysis. The proposed algorithm works really well on the accuracy metrics but use of deep learning based methods can be computationally heavy for small edge devices. In [7], Kadri et al. explains a special statistical technique that can be used to detect DDoS attack based on a metric called CRPS. The paper demonstrates how CRPS can be used for probabilistic forecasting of DDoS attack. The paper also demonstrates how it can be used for real time detection in time series based environments. The proposed algorithm uses a single variable for detection of DDoS attack which might not always be a correct method of detecting DDoS attack in such heterogeneous networks like IoT.

We will try to propose an anomaly detection model based on CRPS for Fog-empowered networks with an attempt to show the importance of multiple features and demonstrate how we can use them to detect anomaly in a given traffic.

IoT works on two layer architecture or a three layer architecture and is vulnerable to various cyber security attacks. One such attack is Distributed denial of services(DDoS). DDoS attack is an attempt to disrupt targeted servers in a malicious way. DDoS attack takes help of some compromised nodes and floods the targeted server with requests ultimately leading to the breakdown of the server. At a higher standard, DDoS attacks clog the destination server with unexpected traffic. DDoS attacks are a little less complicated than any other cyber Security attacks. There are 3 categories to DDoS attack (a) Volume based attacks (b) Protocol attacks (c) Application based attacks. Common types of DDoS attack include SYN flood attack, UDP attack, ICMP attack etc.

Recently, it has been seen that IoT devices have been a major force to drive DDoS attacks. This is a threat that has not been diminished. Security in IoT has been a major talk as it opens up many avenues for attacks to take place. It is anticipated that around 20.4 million devices are due to be deployed by 2020, and it is safe to say that the scale of DDoS attack utilizing this vulnerability could have serious consequences. So, it is important that researchers work on a security technology that is well baked from the start. Some further observation that motivated us to work on DDoS is the use of Multivariate statistical algorithms in detection of DDoS attack. Considering multiple variables rather than single variables can bring out some interesting results.

With respect to the interest and motivation presented earlier our aim with this project is to propose an anomaly detection Model for fog empowered networks. With the help of fog nodes present in the network we can detect anomalies locally. It also eases the task of detecting the malicious node for future research works. We will discuss more about this in further sections. We want to use statistics based algorithms for the purpose of anomaly detection(DDoS Detection). These algorithms are usually used for time series based datasets where packets tend to possess a trend in their incoming pattern. In a statistics based algorithm, to detect the DOS and DDOS attacks, every new traffic network measure is compared to the reference attack free traffic distribution. In simple words trends are analysed and on the basis of the trends detected anomalies are spotted.

In most of the recent works, usually one feature for example : number of data packets per unit time, is used for the detection of DDoS attacks for time series based datasets. Our aim is to use more than one parameter by analysing the dataset which will help in detection of anomalies. The IoT traffic is heterogeneous in nature so using one feature may not be a correct criteria always for judging the anomaly. Therefore, in further sections we will discuss the valuable parameters that can be extracted from the dataset and how we can detect anomalies by implementing statistics based algorithms. Our algorithm uses dimensionality reduction which will be used for extracting important information from features generated and the algorithm is based on Continuous ranked probability score(CRPS) score. CRPS is used in probabilistic forecasting. We will be using python for implementing the algorithm on a dataset and present the output.

The primary contributions of this paper are:

1. Proposed a model for anomaly detection of protocol based attacks.
2. Highlighting the importance of 3 layer architecture in IoT based networks.
3. Mentioned the importance of multiple features while detecting anomalies in IoT traffic and how it can be incorporated in the CRPS metric using dimensionality reduction techniques.

The rest of the paper is organized as follows. In Section 2, the detailed analysis of the concepts used is written. Section 3 explains the component wise working of the proposed anomaly detection model. In Section 4, the evaluation results are discussed with the help of obtained graphs and findings. Section 5 concludes the work by mentioning the key achievements and scope for future works.

2. Detailed analysis

Recently, the source code of a famous mirai malware was released, According to the experts the malware scans the web for IoT devices that have not been secured properly and infect them, as it is easier to hack them. After the devices have been compromised they are used as a part of a botnet that is directed to launch DDoS attacks with malicious intention. All the IoT devices at a global stores its information in the cloud, compromising the IoT devices means hackers can cause discrepancy in the information stored in the cloud which might cause a loss of large information. The problem is that the experts believe that the cyber security attacks on IoT are nowhere stopping here. With lack of security the attacks will increase in a vast number.

Being one of the hottest technologies in recent times, IoT should come with a great responsibility of security. Although it has numerous benefits, these devices are easily hacked which gives it a negative image in the market. IoT should be using new hardware and software which are being generated daily, eradicating the old technologies.

The main challenge with this technology is to detect anomalies in real time and check if the system is affected or not as early as possible since the system is monitoring multiple devices at all times [8,9]. Further examples of Security can be home security, autonomous vehicles security, data management using autonomous systems etc.

2.1. Advantages of fog computing

As discussed in Section 1, Fog Computing includes a 3-layered network architecture where the topmost layer is cloud layer which consists of centralized huge data centres and servers, middle layer is called fog layer where small local servers are present really close to the bottom most layer of sensory nodes to support low latency real time processing. Nodes in the fog layer are called fog nodes. These fog nodes receive data from a small environment and process that data to send desired results with very low propagation time delay. After sending the required response back, the data can then be sent to the cloud servers for further processing and analysis. Usually a group of smart homes or a small network of roads or a production line in a factory can have its own fog server node to collect and process data coming from a bunch of sensory nodes such as cameras, thermostats, pressure gauges etc. [10].

Apart from providing low latency, real-time data processing and distributed network of processing units, addition of fog layer also provides security benefits. Because the data traffic coming from the sensory nodes passes through the fog nodes first, many security checks can be performed to prevent anomalous data or traffic from reaching cloud servers. This mechanism helps in avoiding cyber attacks like DoS or DDoS attacks to a great extent by alarming the network whenever a node tries to flood the server by sending a large amount of packets (SYN packets or Echo reply packets) in a very short amount of time [11]. On getting the alarm, traffic coming from a particular local network under the connection the fog node which raised the alarm can be dumped or blocked to keep the servers safe. The sensors also known as edge nodes are not very secure because of their simple design and application which makes them vulnerable to a lot of cyber attacks. Cyber criminals use these kinds of compromised nodes as bots to launch various kinds of distributed attacks usually on single points of failures in the networks. But the non-centralized nature of the fog empowered networks prevents these edge nodes from getting exposed in the public network and therefore improves security of the network [12].

2.2. Anomaly detection

Anomaly detection can be defined as the process of identifying abnormal or unexpected patterns within a dataset. Abnormal or unexpected patterns are basically just data points which differ from the normal trend of the dataset as a whole. For e.g. in water transportation pipelines, a particular range of pressure is important to be maintained and anything above or below that range can cause critical situations. In this case if the pressure shoots very high suddenly or maybe falls unexpectedly then it can be considered as an abnormal behaviour or an anomaly as compared to a normal scenario. These kinds of anomalies are introduced in the system either by accident or by human mistake. But there are also some cases where this is done by someone intentionally. For e.g. in cyber attacks, an attacker tries to compromise a node or some part of a network which creates anomalous behaviour in network traffic or communication.

As it can be inferred from the above mentioned examples that an anomaly is generally not desired. It is a rare situation but it can still lead to critical scenarios. So there is a need for anomaly detection mechanisms which can detect these abnormal behaviours in the system and warn the concerned authorities to take necessary actions. [13–15] Researchers have proposed many effective anomaly detection techniques based on machine learning, neural networks, statistical analysis etc. These anomaly detection algorithms usually train on unsupervised datasets and try to find the pattern or we can say distribution of the normal trend then any new data point is compared to this pattern that the algorithm has extracted out. If the new data fits in the trend and follows the similar distribution then the algorithm classifies it as normal data otherwise it is classified as an anomaly. The comparison of the data point to the distribution can be based on various metrics [16]. One such metric is called continuous ranked probability score (CRPS) which is used for probabilistic forecasting. We will discuss this metric in detail in next section.

3. Proposed approach

In this work, we are proposing a 3-step anomaly detection framework to detect DDoS attacks in fog networks. Fig. 1 shows the steps involved in the process starting from gathering data from the network traffic to finally detecting the attack. In our approach, we are using a statistical algorithm which works on features extracted from the incoming traffic. These features are first passed in a PCA component before putting the values in the algorithm. This is done because the algorithm used is single variate.

First step of the process is called the data preprocessing step where packets are bundled into time bound windows and then features like number of packets, average time between the packets, number of source IPs etc. can be extracted depending on the network state and requirements. Second step is a Principal Component Analysis(PCA) step where these features are transformed into principal components and the component with the maximum variance along its axis is selected as the main variable to be passed onto the next step.

The final step is the detection of anomalies present in the dataset. This step uses a statistical metric called CRPS which is used to calculate the difference in the distribution of testing data and training data. This difference can indicate the measure of anomaly in the network traffic,

3.1. Data preprocessing

The proposed mechanism deals with time series data which means real-time sequential flow of packets moving from a few source ip addresses to some other destination ip addresses in a network. When these packets pass through the monitoring fog nodes, traffic information is gathered usually for network analysis, security check-ups, traffic behaviour monitoring etc. In our proposed architecture, packet information such as time-stamp, source IP, destination IP, type, packet length etc. is collected. These packets i.e the collected data entries, are then bundled in the groups or we can say windows of 60 s each based on the time-stamps. This is called the Data-Aggregation step. The window size can be changed depending on the network traffic.

This time window is nothing but a hyper-parameter which can be tuned as per the need of the setup in which the algorithm is being tested. But, this tuning needs to be done keeping a few things in mind like the frequency of packets and average amount of packets getting transmitted in the attack free environment before allowing real world data that might allow an attack because if the average data load is high or we can call the throughput of the network is usually high and time window is too small, then it might become too sensitive to the normal increase in the number of packets. Similarly if the throughput is low and the time window is too large then it might miss the sudden impulses or increase in packet frequency which might be a part of a distributed attack.

In developing anomaly detection mechanisms for issues like DDoS attacks, extracting valuable features from the network or the traffic that can be exploited to detect suspicious behaviour of edge nodes or unexpected incoming network traffic is a challenging task. In most of the recently proposed time-series based DDoS detection mechanisms, the number of packets passing through the monitoring point within a fixed time limit or a time window, is used as a feature to detect attacks like TCP-SYN attacks or smurf attacks etc. This can give a rough idea about the network traffic and its behaviour but this feature is not sufficient enough to detect all the hidden patterns in the data. There can be many cases where the packet flow per time window is not exceptionally high but there are a few nodes which are consuming abnormal amount of energy from the network or if in case of DDoS attacks particularly, there can be few instances where some amount of packets were transmitted from a bunch of source node to a target destination node in a very short amount of time or even almost at the same time. Now this amount of packet flow can be detected as slightly anomalous if the total packet count within the time window is significantly large otherwise it will be considered as normal traffic. Whereas these impulses of packets from a group of different networks can accumulate and flood the target node.

Therefore, anomaly detection mechanisms based on multiple variables are more reliable and accurate. In this work, Along with the number of packets passing in the network within a time window, which is a common feature used by many researchers, the average time between the subsequent packers within the time window is used as an another parameter to tackle the above discussed issue of many nodes flooding the target by sending very high frequency of packets in a very short period of time. This another feature is important because this provides a new perspective to the situation. Suppose there is a window of say 60 s, in one case there are 100 packets passing through the network within these 60 s and the flow of packets is uniformly distributed throughout this time window, and in second case, there are no packets within first 30 s and then suddenly there are 100 packets

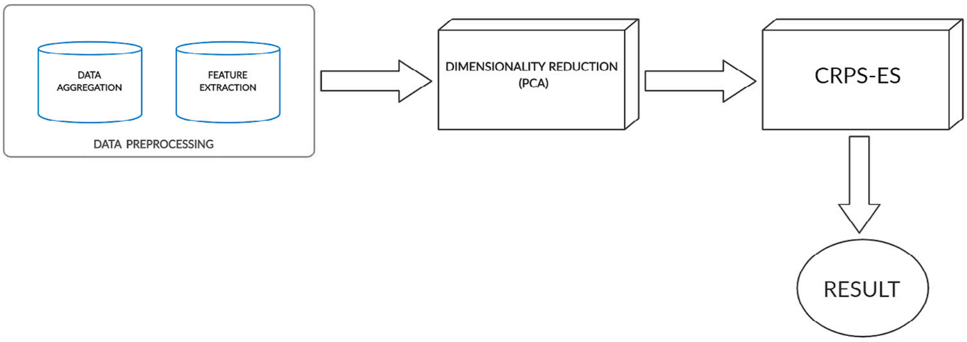

Fig. 1. The Proposed Architecture.

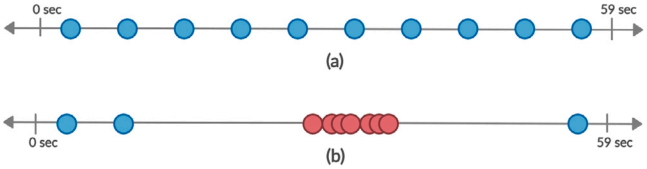

Fig. 2. Difference in the average time between the packets given the number of packets is same.

within 1 s, from 30th to the 31st second and then again no packets till 60th second. Both these cases are showing a very different behaviour though the number of packets is the same. The only difference is the average time between the packets that can project that the flow in the second case is anomalous (see Fig. 2).

The calculation of both the above discussed features for every time window is performed in the Feature Extraction step which also concludes the Data Preprocessing. After that a trend based on the combined effect of the extracted features is calculated in the dimensionality reduction component which uses principal component analysis or PCA. We have discussed more about this in the next section.

3.2. Principal component analysis

Dimensionality reduction includes transformation of data from higher dimension to a lower dimension where the lower dimension features extract useful information from the original data. In our case we required dimensionality reduction because there was a need to transform multi-dimensional feature data to a single dimensional data as CRPS accepts data with single feature. This way we combine the effect of multiple features and the most relevant information out of all the features is passed to the statistical algorithm. PCA was a perfect fit for our algorithm because it tries to extract information on the basis of maximum variance as the basis [17]. Selecting information on the basis of maximum variance will help us to extract varying information while ignoring the information not required.

As discussed , we are using two variables for our algorithm. We have used PCA to convert the two variables into a single variable to meet our requirements. Time series based anomaly detection algorithms usually consider a single variable but using PCA we can produce a single variable which is a combination of variations along different metrics which can obviously be considered a better choice for anomaly detection considering the fact that the IoT traffic is heterogeneous in nature.

There are certain conditions that have to be met to apply PCA. So it is very important to show the correlation between different variables that we have considered in the previous section. We will be using graphical results to explain the correlation in the Evaluation section.

3.3. Continuous ranked probability score

CRPS or continuous rank probability score is a statistical metric which is commonly used in probabilistic forecasting models. In our work, we have selected this metric because CRPS can be used to quantify the dissimilarity between a new observation and the attack free traffic distribution [18].

This makes CRPS a better suited algorithm for real time detection as compared to other similar metrics like KL divergence or even X2(chi square) which requires the whole data beforehand to compute distributions of training and testing data [13]. At the time of testing, the difference between the CRPS values of the incoming data points and the attack free normal data can identify the abnormal behaviour of the network traffic [7].

Let X be the random variable. Let F be the cumulative distribution function (CDF) of X, such as $F(y) = P[X \leq y]$. Let x be the observation, and F the CDF associated with an empirical probabilistic forecast. The CRPS between x and F is defined as Eq. Eq. (1) in Algorithm 1, where $1(x)$ is the Heaviside step function and denotes a step function along the real line that attains, first, the value of 1 if the real argument is positive or zero, and second, the value of 0 otherwise.

The CRPS is expressed in the same unit as the observed variable. Eq. Eq. (1) represents the measure of the difference between the underlying distribution of the given attack free data and the incoming observation data. For our case we know that the attack free data is normally distributed. Now, when we substitute the function F as a normal distribution we get Eq. Eq. (2). Therefore, Eq. Eq. (2) is just an extension of Eq. Eq. (1).

To enhance the detection efficiency of CRPS we apply Exponential smoothing to it (CRPS-ES). This is done for the inclusion of previous and current information in the decision process which helps it in uncovering even small anomalies and makes the statistic less sensitive towards noise.

Usually the distribution of data is assumed to be gaussian but here we try to figure out a more realistic distribution for the data using the Kernel Density Estimation (KDE) [19]. CRPS-ES considers a non-parametric decision threshold which is computed by analysing the flow of the underlying distribution. The major plus point of CRPS-ES metric is that it considers the past data in the detection which makes it sensitive to apparent attacks. [20] It is more suitable for real time detection as the new traffic that might contain anomalies is compared to the attack free traffic (see Fig. 3).

4. Evaluation

We have used a very popular dataset used for Wireless Sensor Networks to test our algorithm which is DARPA99 dataset. It is an anomaly detection evaluation dataset generated in 1999 by Lincoln Laboratory at MIT.

This dataset consists of data packets passing through a network over a period of 5 weeks. The first 3 weeks are a part of the training

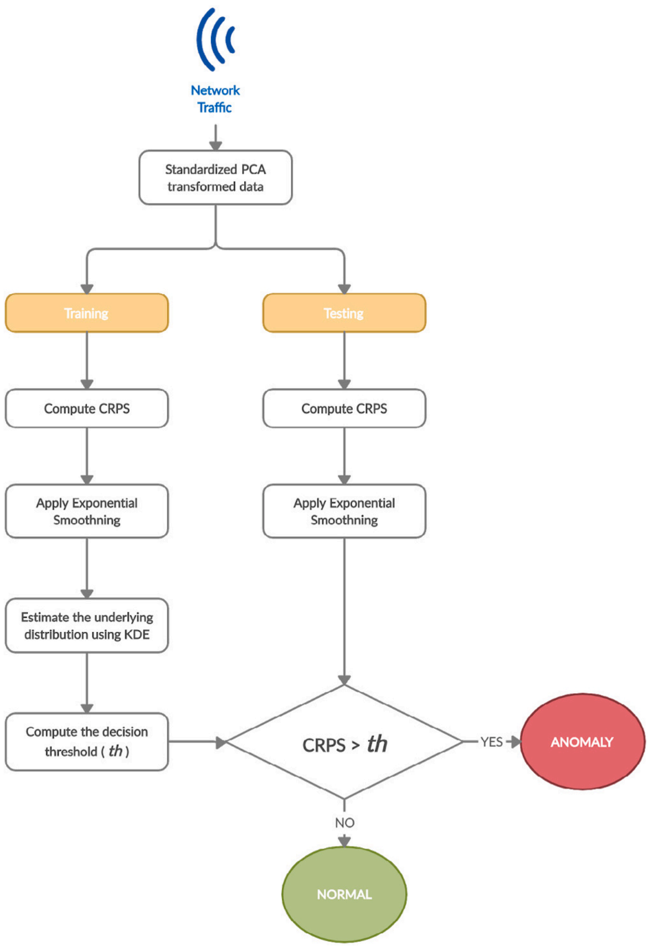
Network Traffic
Standardized PCA transformed data
Training
Testing
Compute CRPS
Compute CRPS
Apply Exponential Smoothning
Apply Exponential Smoothning
Estimate the underlying distribution using KDE
Compute the decision threshold (th)
CRPS > th
YES
ANOMALY
NO
NORMAL

Fig. 3. Flowchart for Anomaly Detection.

dataset with various labelled information and many different types of evaluations and observations most of which are out of the scope of this work. The last 2 weeks are for testing where the many cyber attacks like TCP-SYN, ICMP etc. were launched on the network under observation.

Wireshark is a tool used for analysing data packets sent from one IP address to another. We have used Wireshark to extract out important information from the DARPA 99 dataset and use it for our experiment purpose.

As depicted in above snapshot packets in DARPA 99 contain a timestamp, Source IP, destination IP, Protocol used, Length of the message and some information about. For a particular type of attack the packets with the same type of protocol used are filtered out. For e.g. For, TCP SYN attack packets with TCP protocol are filtered out. Wireshark provides an inline command facility to filter out packets.

Here for our experiment we have used Week 1 Day 1 as an attack free traffic as there is no attack on Day 1 Week 1. For detection purposes(testing data) we have used Week 5 Day 1 as it contains a TCP SYN attack initiated once for a duration of 6 min 51 s and for ICMP Week 4 Day 1 is used it has 2 attacks for 1s each (see Figs. 4–16).

4.1. Observations for training data for TCP-SYN attack

See Figs. 4–7.

4.2. Results for testing data for TCP-SYN attack

See Figs. 8–10.

Algorithm 1: Training Algorithm

```
1: In this phase we will have a preprocessed data with PCA applied
   to it. We will use the CRPS metric on this data. CRPS is helpful in
   quantifying the deviations from attack free data which help us in
   detecting anomalies. For a observation x, the CRPS value is
   calculated as
   $CRPS(F,x) = \int_{-\infty}^{\infty} (F(y) - 1(y \geq x))^2 dy$    (1)
   where 1(y ≥ x) :
   $1(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$    (2)
   It should be noted that when the traffic is attack free its
   distribution is gaussian. So, the CRPS for function F as gaussian
   will be :
   $CRPS(N(\mu,\sigma^2),x) = \sigma[\frac{x-\sigma}{\sigma}(2\phi(\frac{x-\sigma}{\sigma})-1)+2\phi(\frac{x-\sigma}{\sigma})-1/\sqrt{\pi}]$    (3)
2: Now the exponential smoothing is applied. This is done for the
   inclusion of previous and current information in the decision
   process which helps it in uncovering even small anomalies and
   makes the statistic less sensitive towards noise. The mathematical
   formula is calculated as
   $z_t^{CRPS} = vd_t + (1-v)z_{t-1}^{CRPS}$    (4)
   Where, $z_t$ is the calculated value of the current data point, $z_{t-1}$ is
   the calculated value of previous data , $d_t$ is the observation of the
   current data point and $v$ is the forgetting parameter.
3: At this step, we have the calculated CRPS-ES values for the
   training data. Now, a non-parametric threshold (th) value needs
   to be calculated to detect the anomalies in the testing phase. This
   threshold should be able to accurately justify the results i.e. if a
   value is crossing the threshold, it must differ from the underlying
   distribution of the training CRPS-ES statistic. This distribution is
   calculated using Kernel Density Estimation. The threshold is
   calculated as the $(1-\alpha)$th quantile of the calculated distribution.
```

Algorithm 2: Testing Algorithm

```
1: In this phase we will have a preprocessed data with PCA applied
   to it similar to the training phase. CRPS is applied similar to the
   training phase.
2: Then Exponential smoothing is applied and CRPS_ES value is
   obtained.
3: The output of step 2 is compared to the threshold obtained from
   the training phase.
4: if CRPS_ES > th then
5:     Anomaly Present
6: else
7:     Normal Traffic
8: end if
```

4.3. Observations for training data for ICMP attack

See Figs. 11–13.

4.4. Results for testing data for ICMP attack

See Figs. 14–16.

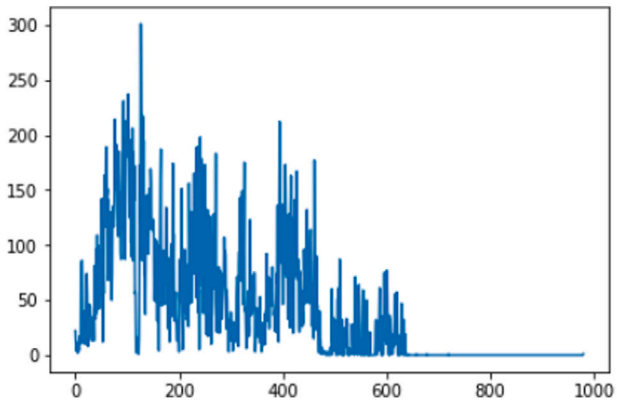

Fig. 4. First Feature: the number of packets passing through the network per 75 s window. Here, x-axis: the index of windows and y-axis: the first feature values.

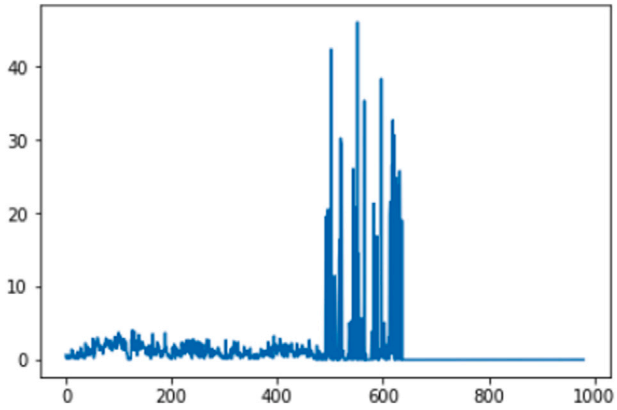

Fig. 5. Second Feature: an inverse function of average time between the packets within the 75 s window. Here, x-axis: the index of windows and y-axis: the second feature values.

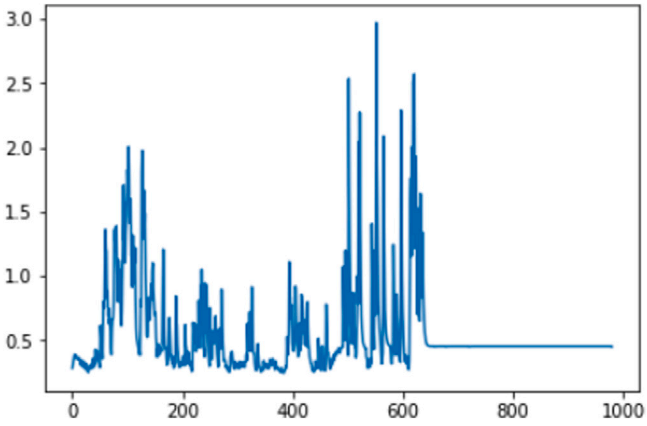

Fig. 6. CRPS-ES values for training data after applying PCA. Here, x-axis: the index of windows and y-axis: the CRPS-ES values.

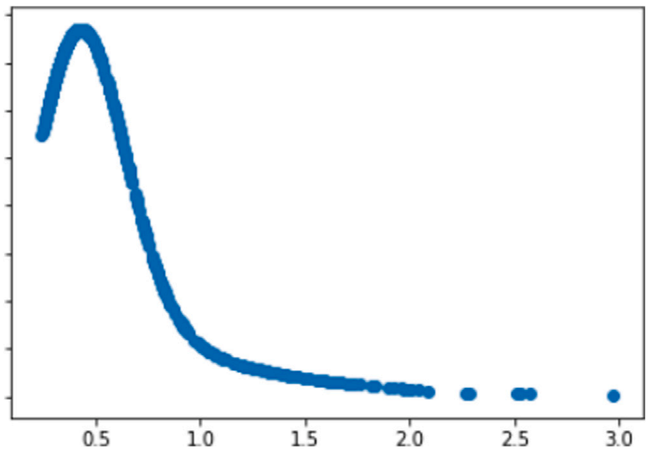

Fig. 7. Distribution of the CRPS-ES calculated using KDE.

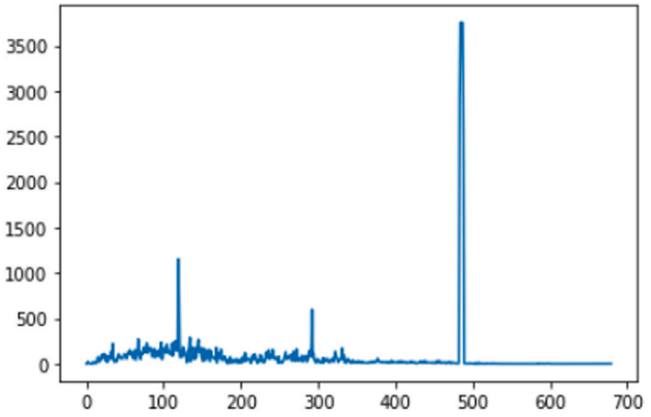

Fig. 8. First Feature: the number of packets passing through the network per 75 s window. Here, x-axis: the index of windows and y-axis: the first feature values.

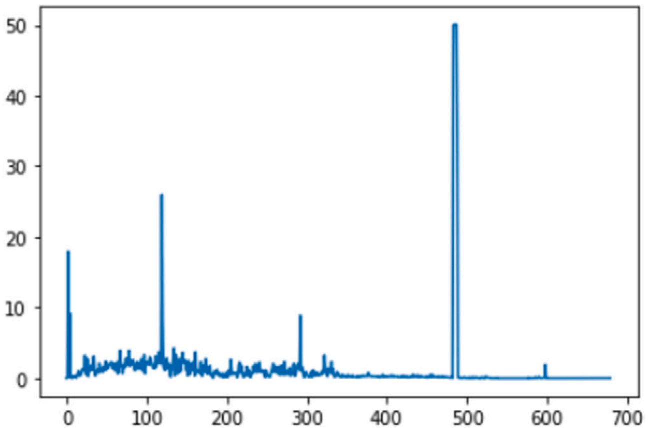

Fig. 9. Second Feature: an inverse function of average time between the packets within the 75 s window. Here, x-axis: the index of windows and y-axis: the second feature values.

5. Conclusion

We were able to detect all the TCP-SYN and ICMP attacks present in the dataset used using fast statistical algorithms as compared to deep learning mechanisms which are computationally very expensive and are not feasible for edge and fog nodes. Other protocol based attacks include UDP attacks. This algorithm can work on most of the protocol based attacks. The common idea behind all protocol based attacks remains the same, that is analysing the incoming traffic based on some window size, reducing its dimensions and applying the CRPS-ES metric. We were also able to show the importance of fog computing in the anomaly detection frameworks as it provides local monitoring of the network traffic which can help us in detection of various attacks at the lowest level. Our aim here was to highlight the importance of multiple features. in time series based algorithms. We have demonstrated why

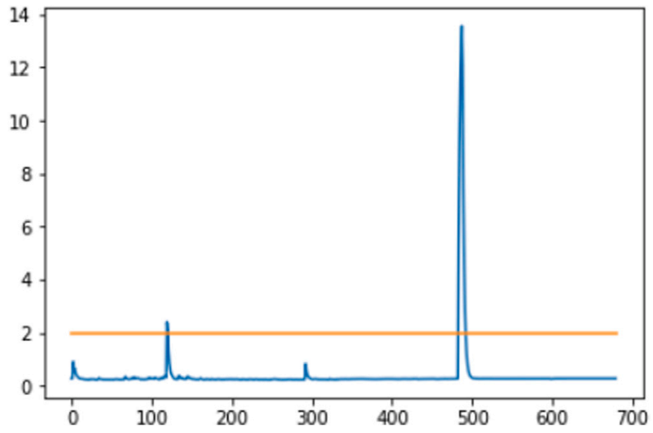

Fig. 10. CRPS-ES values for training data after applying PCA. Here, x-axis: the index of windows and y-axis: the CRPS-ES values. The orange line represents the threshold CRPS-ES value (th = 1.97, $(1 - \alpha) = 0.99$). The peaks in the figure are the attack points with CRPS-ES values > th. It can be observed in the figure that all the attacks are successfully detected by the proposed algorithm.

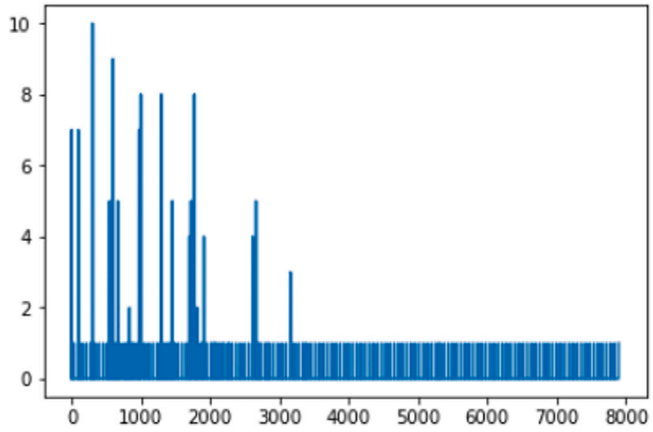

Fig. 11. First Feature: the number of packets passing through the network per 75 s window. Here, x-axis: the index of windows and y-axis: the first feature values.

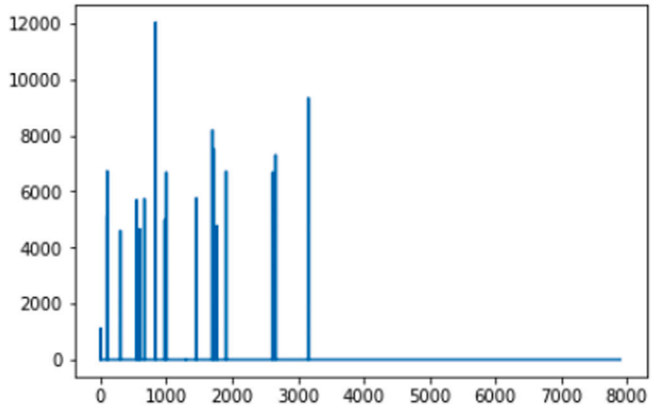

Fig. 12. Second Feature: an inverse function of average time between the packets within the 75 s window. Here, x-axis: the index of windows and y-axis: the second feature values.

we should use multiple features and how we can extract multiple features from the network. Use of multiple features plays an important role in detecting hidden attacks and exposing various unknown trends present in the data.

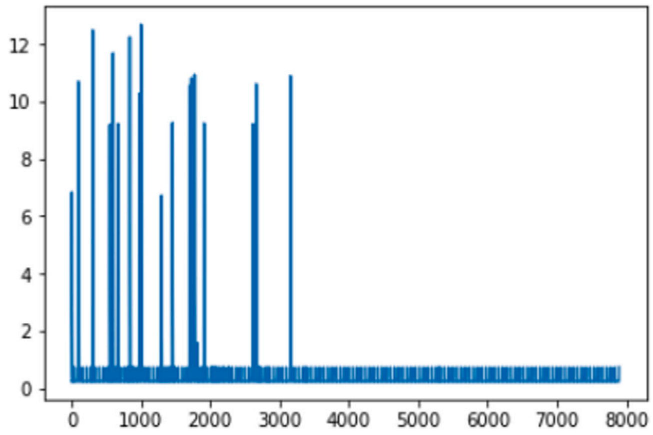

Fig. 13. CRPS-ES values for training data after applying PCA. Here, x-axis: the index of windows and y-axis: the CRPS-ES values.

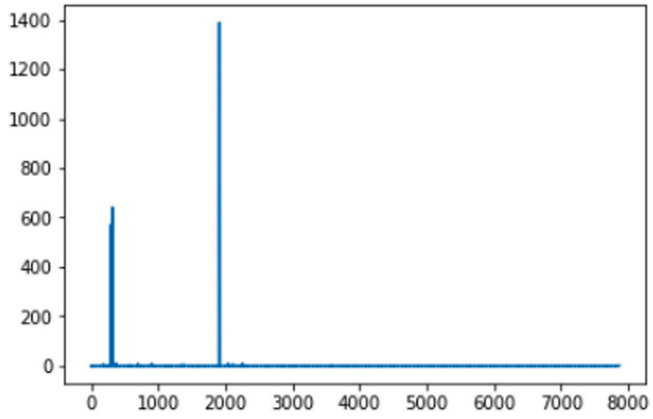

Fig. 14. First Feature: the number of packets passing through the network per 75 s window. Here, x-axis: the index of windows and y-axis: the first feature values.

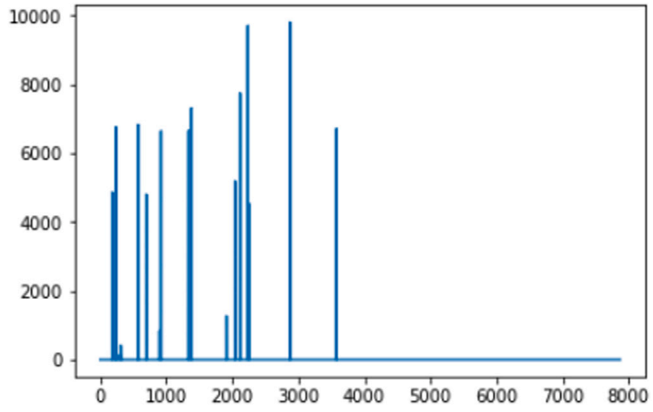

Fig. 15. Second Feature: an inverse function of average time between the packets within the 75 s window. Here, x-axis: the index of windows and y-axis: the second feature values.

But, CRPS is a single variable algorithm which is a negative point of the algorithm, though there exists a multi variable CRPS statistical method but it is not very easy to implement. Therefore , an approach was devised to incorporate multiple variables in the CRPS algorithm by using dimensionality reduction technique. But this leads to some loss of information as well which is a shortcoming of the model as there is a limit on the number of features that can be reduced to a single variable.

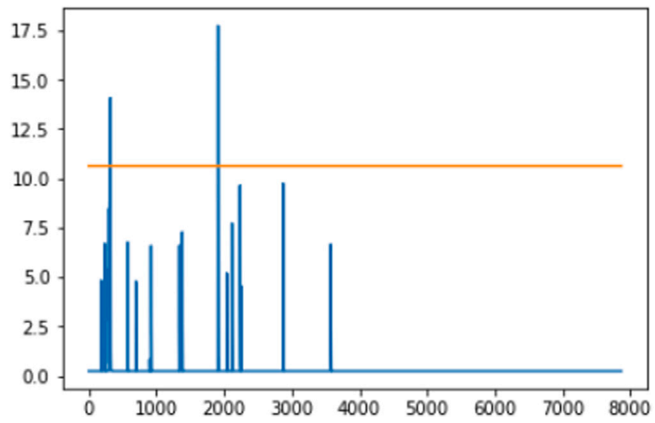

Fig. 16. Here, x-axis: the index of windows and y-axis: the CRPS-ES values. The orange line represents the threshold CRPS-ES value (th = 10.62, $(1-\alpha) = 0.99$). The peaks in the figure are the attack points with CRPS-ES values > th. It can be observed in the figure that all the attacks are successfully detected by the proposed algorithm.

In terms of objective comparative studies, previous models were also suitable as researchers were able to detect obvious attacks present in the data but if we consider niche situations and scenarios then it is clearly visible through the observations and results that our model is capturing more information from the network traffic and is more robust.

Our approach is modular as well because hyper-parameters can be tweaked and various set of features can be used depending on the use case and state of the network.

In future works, this architecture can be made more accurate and useful by,

1. Including more network parameters such as energy consumption, number of source IPs etc., as features in the anomaly detection model.
2. Using multivariate CRPS components because this can increase the capability of extracting network features and using them in analysing the data trends by manifolds.
3. Identification of the malicious nodes present in the network. In this work we have detected the anomaly present in the data but finding the source of the anomaly data is still a very challenging task that we would like to tackle in our future works.
4. Exploring the capabilities of Fog computing and figuring out how we can leverage it to enhance the efficiency of our model to the maximum.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Bashar Ahmad Khalaf, Salama A. Mostafa, Aida Mustapha, Mazin Abed Mohammed, Moamin A. Mahmoud, Bander Ali Saleh Al-Rimy, Shukor Abd Razak, Mohamed Elhoseny, Adam Marks, An adaptive protection of flooding attacks model for complex network environments, Secur. Commun. Netw. 2021 (2021) 17, http://dx.doi.org/10.1155/2021/5542919, Article ID 5542919.

[2] Adnan Helmi A. zizan, Salama A. Mostafa, Aida Mustapha, Cik Feresa Mohd Foozy, Mohd Helmy Abd Wahab, Mazin Abed Mohammed, Bashar Ahmad Khalaf, A machine learning approach for improving the performance of network intrusion detection systems, Ann. Emerg. Technol. Comput. (AETiC) 5 (5) (2021) 201–208, http://dx.doi.org/10.33166/AETiC.2021.05.025, Print ISSN: 2516-0281, Online ISSN: 2516-029X, Published by International Association of Educators and Researchers (IAER), Available: http://aetic.theiaer.org/archive/v5/v5n5/p25.html.

[3] Shafiq Muhammad, Zhihong Tian, Ali Bashir, Alireza Jolfaei, Data mining and machine learning methods for sustainable smart cities traffic classification: A survey, Sustainable Cities Soc. 60 (2020) http://dx.doi.org/10.1016/j.scs.2020.102177.

[4] R.M.A. Ujjan, Z. Pervez, K. Dahal, et al., Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN, Future Gener. Comput. Syst. (2019) http://dx.doi.org/10.1016/j.future.2019.10.015.

[5] Yonghao Gu, Yongfei Wang, Zhen Yang, Fei Xiong, Yimu Gao, Multiple-features-based semi supervised clustering DDoS detection method, Math. Probl. Eng. 2017 (2017) 10, http://dx.doi.org/10.1155/2017/5202836, Article ID 5202836.

[6] S. Sumathi, N. Karthikeyan, Detection of distributed denial of service using deep learning neural networks, J. Ambient Intell. Hum. Comput. (2020) http://dx.doi.org/10.1007/s12652-020-02144-2.

[7] F. Harrou, B. Bouhaddou, Y. Sun, B. Kadri, Detecting cyberattacks using a CRPS-based monitoring approach, in: 2018 IEEE Symposium Series on Computational Intelligence, SSCI, 2018, Available: http://dx.doi.org/10.1109/SSCI.2018.8628797.

[8] S. Madakam, R. Ramaswamy, S. Tripathi, Internet of things (IoT): A literature review, J. Comput. Commun. 3 (2015) 164–173, http://dx.doi.org/10.4236/jcc.2015.35021.

[9] D. Myridakis, G. Spathoulas, A. Kakarountas, D. Schoinianakis, J. Lueken, Anomaly detection in IoT devices via monitoring of supply current, in: 2018 IEEE 8th International Conference on Consumer Electronics, Berlin (ICCE-Berlin), 2018, pp. 1–4, http://dx.doi.org/10.1109/ICCE-Berlin.2018.8576178.

[10] A.A. Mutlag, M. Khanapi Abd Ghani, M.A. Mohammed, M.S. Maashi, O. Mohd, S.A. Mostafa, K.H. Abdulkareem, G. Marques, I. de la Torre Díez, MAFC: Multi-agent fog computing model for healthcare critical tasks management, Sensors 20 (2020) 1853, http://dx.doi.org/10.3390/s20071853.

[11] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, B. Sikdar, A survey on IoT security: Application areas, security threats, and solution architectures, IEEE Access 7 (2019) 82721–82743, http://dx.doi.org/10.1109/ACCESS.2019.2924045.

[12] Shanhe Yi, Cheng Li, Qun Li, A survey of fog computing: Concepts, applications and issues, in: Proceedings of the 2015 Workshop on Mobile Big Data, Mobidata '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 37–42, http://dx.doi.org/10.1145/2757384.2757397.

[13] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, H. Ming, AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning, in: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC, 2019, pp. 0305–0310, http://dx.doi.org/10.1109/CCWC.2019.8666450.

[14] Zhongguo Yang, Irshad Ahmed Abbasi, Elfatih Elmubarak Mustafa, Sikandar Ali, Mingzhu Zhang, An anomaly detection algorithm selection service for IoT stream data based on tsfresh tool and genetic algorithm, Secur. Commun. Netw. 2021 (2021) 10, http://dx.doi.org/10.1155/2021/6677027, Article ID 6677027.

[15] A.A. Cook, G. Mısırlı, Z. Fan, Anomaly detection for IoT time-series data: A survey, IEEE Internet Things J. 7 (7) (2020) 6481–6494, http://dx.doi.org/10.1109/JIOT.2019.2958185.

[16] G. Spanos, K.M. Giannoutakis, K. Votis, D. Tzovaras, Combining statistical and machine learning techniques in IoT anomaly detection for smart homes, in: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD, 2019, pp. 1–6, http://dx.doi.org/10.1109/CAMAD.2019.8858490.

[17] T. Jolliffe Ian, Cadima Jorge, Principal component analysis: a review and recent developments, Phil. Trans. R. Soc. A 374 (2016) http://dx.doi.org/10.1098/rsta.2015.0202, 2015020220150202.

[18] B.A. Khalaf, S.A. Mostafa, A. Mustapha, M.A. Mohammed, W.M. Abduallah, Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods, IEEE Access 7 (2019) 51691–51713, http://dx.doi.org/10.1109/ACCESS.2019.2908998.

[19] Weglarczyk Stanislaw, Kernel density estimation and its application, in: ITM Web of Conferences, Vol. 23, 2018, p. 00037, http://dx.doi.org/10.1051/itmconf/20182300037.

[20] Tilmann Gneiting, Adrian E. Raftery, Strictly proper scoring rules, prediction, and estimation, J. Amer. Statist. Assoc. 102 (477) (2007) 359–378, http://dx.doi.org/10.1198/016214506000001437.

Deepak Kumar Sharma is working as an Assistant Professor in the Department of Information Technology, Netaji Subhas University of Technology (Formerly NSIT), Dwarka, New Delhi, India. He obtained his Ph.D. in Computer Engineering from University of Delhi, India in 2016. His research interests include opportunistic networks, wireless ad hoc and sensor networks, Software Defined Networks and IoT Networks. He has over 15 years of experience in Academics. He has published various research papers in reputed international journals like ETT Wiley, IEEE Systems Journal, Computer Communication Elsevier, IJCS Wiley etc. and conferences of repute like IEEE AINA, GLOBECOM etc. He has also authored various book chapters in edited books of IET, Wiley, Springer, Elsevier etc. He is also a reviewer of various reputed journals like ETT Wiley, AIHC Springer, IJCS Wiley etc.

Tarun Dhankhar is currently pursuing is undergraduate Bachelor's degree in Information Technology from Netaji Subhas Institute of Technology, New Delhi, India. His research interests are in the areas related to Fog Computing, the Internet of Things and the application of Statistical techniques in these areas. His specific interests are the application of Information Security in these research areas. Previously He has worked in the fields of Machine Learning and Deep Learning as well.

Gaurav Agrawal is currently pursuing his undergraduate Bachelor's degree in Information Technology from Netaji Subhas Institute of Technology, New Delhi, India. His research interests are in the areas related to the Internet of Things, Fog Computing and the application of Statistical techniques in these areas. His specific interests include topics like Machine Learning, Deep Learning.

Satish Kumar Singh is working as an Assistant Professor in the Department of Information Technology, Netaji Subhas University of Technology (Formerly NSIT), Dwarka, New Delhi, India. He is currently pursuing his Ph.D. in Information Technology from Netaji Subhas University of Technology, Delhi, India. He has obtained M.Tech Integrated degree in Information Technology from University School of Information Technology, GGSIPU, New Delhi. His research interests include Cloud Computing, Fog Computing, Wireless ad hoc and sensor networks, Software Defined Networks and IoT. He has over 10 years of experience in Academics. He has published various research papers in conferences of repute like IEEE, etc.

Deepak Gupta received a B.Tech. degree in 2006 from the Guru Gobind Singh Indraprastha University, India. He received M.E . degree in 2010 from Delhi Technological University, India and Ph. D. degree in 2017 from Dr. APJ Abdul Kalam Technical University, India. He has completed his Post-Doc from Inatel, Brazil. With 13 years of rich expertise in teaching and two years in the industry; he focuses on rational and practical learning. He has contributed massive literature in the fields of Intelligent Data Analysis, BioMedical Engineering, Artificial Intelligence, and Soft Computing. He has served as Editor-in-Chief, Guest Editor, Associate Editor in SCI and various other reputed journals (IEEE, Elsevier, Springer, & Wiley). He has actively been an organizing end of various reputed International conferences. He has authored/edited 50 books with National/International level publishers (IEEE, Elsevier, Springer, Wiley, Katson). He has published 186 scientific research publications in reputed International Journals and Conferences including 99 SCI Indexed Journals of IEEE, Elsevier, Springer, Wiley and many more. He has also filled/published 4 patents.

Jamel Nebhen received the M.Sc. in Microelectronics from the National Engineering School of Sfax, Tunisia in 2007, and the Ph.D. degrees from the Aix-Marseille University, France, in 2012, all in Microelectronics. From 2012 to 2018, he worked as a Postdoctoral Researcher in France in LIRMM-Lab Montpellier, IM2NP-Lab Marseille, ISEP Paris, LE2I-Lab Dijon, Lab-Sticc Telecom Bretagne Brest, and IEMN-Lab Lille. Since 2019, he joined the Prince Sattam bin Abdulaziz University in Alkharj, Saudi Arabia, as an Assistant Professor. His research interests are mainly in the design of analog and RF integrated circuits, IoT, biomedical circuit, and sensors instrumentation.

Imran Razzak is currently Sr. Lecturer at School of Information Technology at Deakin University, Australia. Imran's research interest is machine learning and data analytics in general, particularly in healthcare industry. He is a passionate health informatician who wants to make the healthcare industry a better place through informatics. He has published more than 70 refereed research publications in international journals and conferences. He is an editorial board member of many reputable international journals as well as session co-chair, session chair and TPC member of dozens of conferences. He is also working as a consultant in various projects involving deep learning models in Big Data, IoT, medical Imaging and BCI applications.