European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 07, Issue 06, 2020

Device Indentification In Network Traffic Using Artificial Intelligence Based Capsule Networks

Dr. ThangaMariappan L[1], Dr. Lakshminarayanan R[2], Dr. H.Azath[3]

Associate Professor, School of Computing, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, [1,2] Assistant Professor, Department of CSE VIT – Bhopal[3]

Abstract:

In recent decade, the security threats poses a high risk in an organization, which is associated with the proliferation of IoT devicesand increasing organizational assets. This ensures that the organization is unaware of the IoT devices connection with their own network. In such cases, security and integrity of network might pose a serious security threat to the network communications. In this paper, capsule network, which is an improved version of Convolutional Neural Network (CNN) is used to monitor the network traffic to identify accurately the trusted devices connected to the home network. Inadequacy of CNN in identifying the IoT devices during its communication in the network has made the present research to choose Capsule Networks (CapsNet) for device identification. Capsule network carries out the operation in an iterative manner in order to attain improved classification of IoT devices. The activation function used in the capsule network is a squash function that normalizes the magnitude of vector rather than the conventional usage of scalar elements. The outputs of activation function helps to find the trusted IoT devices through different capsules, which are formally trained using various concepts. The capsule network performs the identification of IoT devices and classifies the trusted and non-trusted devices based on the labeled network traffic data. The simulation is performed by the computation of collected labeled network data from IoT associated network.

Keywords:

Artificial Intelligence, Internet of Things, Capsule network

1. INTRODUCTION

The term “internet of Things” (IoT) is a keyword of a broad space for distributed devices with embedded identification, sensing and/or actuation abilities [1], which encompasses various aspects related to the extension of the Internet and the web into the physical world. Security and governance problems stemming from an increasing number of IoT-enabled organizational assets are among the challenges IoT poses for the organizations. In future, companies may not know exactly what IoT device is connected to their network, a situation that threatens network security and integrity.

Currently, the IT infrastructure in areas such as electromagnetic grids, financial data systems and emergency communication systems is comprised of computer networks. It is critical for the economies and safety of our nation to protect the networks from malicious intrusions. In software applications used for cyber-attacks, vulnerabilities are regularly discovered. Management of a corporate safety risk is more an art than a science at the present time. Instead of relying on objective measures, System Administrators are acting with instinct and experience to guide and justify decisions [2].

The security provided by various network configurations should be measured to improve the safety of business networks. Our research aimed to develop a standard computer network safety measurement model. A standard model allows us to reply to questions like “Are we safer than yesterday?”or“How is one network security setup compared to another?”The standard network security measurement

model will also bring users, vendors and researchers together to assess network security methodologies and products [2].

Initially, the proposed method uses a series of steps that helps in classification of trusted and untrusted IoT device connection in an enterprise network. CapsNet iteratively improves the classification of IoT devices and accurately identifies the trusted IoT devices based on labelled data. This approach is made flexible and generic through a series of steps that includes improved feature extraction using Bat Algorithm. The feature extraction using high level network statistics like average time to live (TTL) and ratio between incoming bytes and outgoing bytes [3, 4, 5]. The extraction and labelling of proper features helps in classification of devices with better accuracy in a reduced execution time.

The main contribution of the work involves the classification of the trusted and non-trusted IoT devices in a network using a deep learning model. The proposed framework is designed to analyse the network traffic to classify and identify untrusted IoT device crawling within an enterprise network. The traffic analysis with property of HTTP packet, Bat feature extraction and CapsNet[6-9] classification helps in accurate classification of IoT device in the network.

The outline of the paper is given as follows: Section 2 provides the details of the existing works. Section 3 discusses the proposed classification method. Section 4 evaluates the proposed method with other existing works. Section 5 concludes the paper with possible directions for future scope.

2. Related works

Singhal, A., &Ou, X. (2017) [2] introduced a security risk analysis model and methodology for corporate networks using probabilistic attack graphs. The model announces a graph of attack with known vulnerabilities and exploitation probabilities. By spreading the likelihood of exploitation via the attack graph, a metric is calculated which quantifies the general safety risk of business networks. This methodology can be used to assess and improve the safety risk of company systems.

In order to correctly identify network-connected IoT devices, Meidan, Y. et al. (2017) [10] applies machine learning algorithms on Net Traffic Data. This method collect and mark network traffic data from 9 different IoT devices to train and evaluate the classifier. We have been trained in a multi-stage meta-classification by means of supervised learning; in the first stage, the classifier can distinguish between IoT and non-IoT traffic systems. Each IoT device is linked to a specific IoT device class in the second phase.

Iglesias, F., &Zseby, T. (2015) [11] address network-based anomaly detection feature selection problem. With filters and stepwise regression wrappers, the study propose a multi-step feature selection method. The method is based on 41 commonly used traffic functions presented in a variety of commonly used traffic information sets. We could reduce original feature vectors from 41 to only 16 features using our combined feature selection method.

The multi-stage outlier approach for detecting network-wide abnormalities was presented by Bhuyan, M. H., et al. (2016) [12]. During clustering and anomaly detection, we identify a subset of traffic functions. We use the following modules to support the identification of outer network anomalies: Mutual information and a generalized entropy-based selection method for selecting a relevant non-redundant subset of characteristics, a tree-based clustering technique to generate a set of benchmarks and an outer scoring function to detect anomalies in incoming network traffic.

There are very few studies used to analyse the network anomalies based on network traffic, however only one study has reported to study the IoT network using machine learning [13, 14]. Hence, we improve this method by the application of a deep learning model on IoT device network identification.

3. Proposed method

This research provides an improved way for the classification of trusted IoT devices connected in an enterprise network using network traffic analysis. Specifically, the study focus on using high level traffic data to accurately classify the trusted and non-trusted IoT devices. The proposed method

initially collects the high level network data through monitoring the enterprise network and then pre-process it, followed by feature extraction, selection using a swarm optimisation[15] technique called Bat algorithm [16] [17]. It feature extraction accurately selects the network features and it is then labeled to ease the process of classification using CapsNet. The series of operation prior to classification improves the reliability of the system. The architecture of which is given in Figure 1(a).

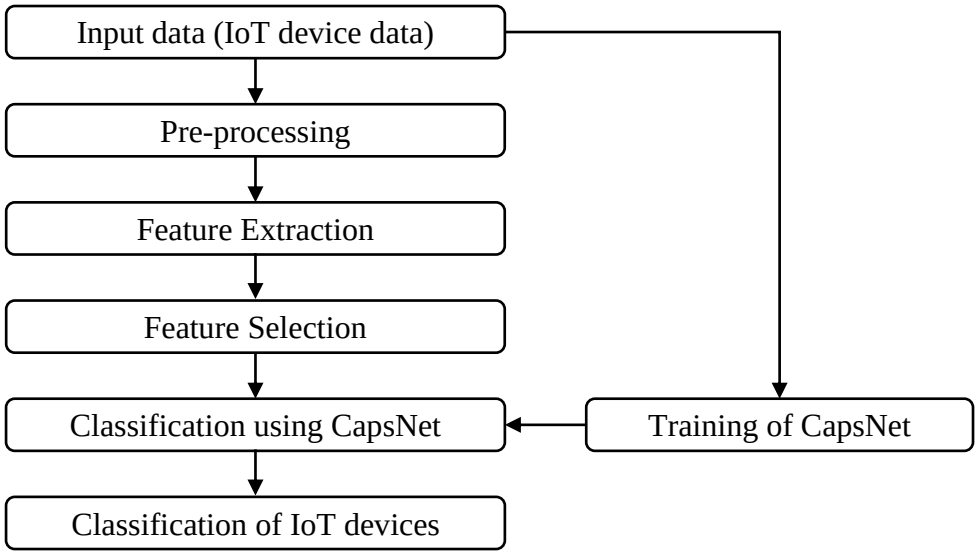

Figure 1(a): Architecture of CapsNet for IoT device classification

3.1. Data Acquisition

To promptand to evaluate the proposed model, the traffic data is collected across the network via devices that falls under two different categories, which includes IoT devices (Baby Monitor, Motion Sensor, Printer, Refrigerator, Security Camera, Socket, Thermostat, TV, Smartwatch) and non-IoT devices (PC, Laptop and Smartphone). As usual these devices are connected to a Wi-Fi access point and the recorded network traffic data (*.pcap files) is used for additional analysis.

3.2. Feature Extraction.

The TCP packets (Packet Format) are converted to unique 4-tuples or sessions (composed of port numbers and IP addresses from SYN to FIN). Each The unique 4-tuples or sessions has a series of network features, application and transport layers are represented and enriched with the GeoIP and Alexa Rank datasets.

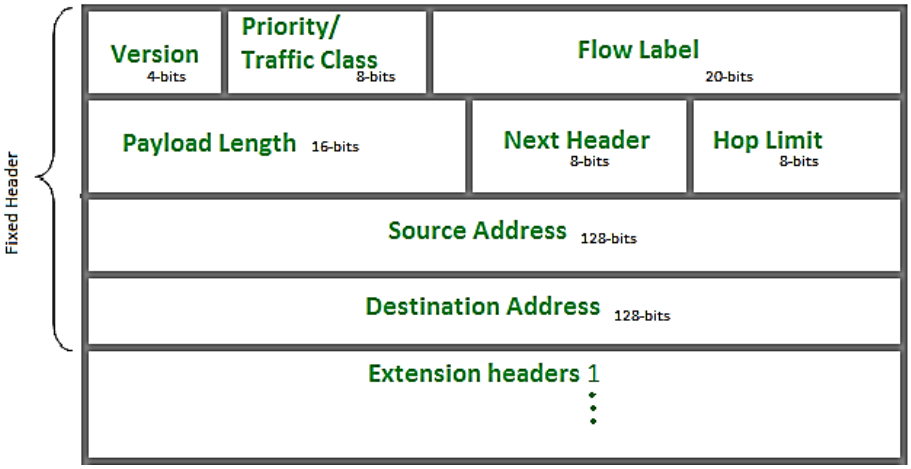

Figure 1(b). Packet Format

3.3. Feature Selection

Feature selection is suggested to select the most important feature subset to reduce the error rate. The bat algorithm provides the most relevant input data for data classification to achieve the exact result.

A metaheuristic enhancement algorithm is the bat algorithm inspirited from bats. In the areas of feature selection, the bat algorithm is applied, where it operates on microbatscholocation behaviour with capricious beating ante of discharge and loudness. The optimisation of the microbats echolocation can be synthesized as:

- The speed v_i at position (solution) $x_i$flies arbitrarily at each exclusive bat at a different speed or wavelength and loudness A_i.

- It changes the occurrence, pulses, loudness and emissions r by exploration and exploitation of the features.

- A random walk in an area intensifies the search. A collection of the best solution goes on until certain stops are met.

This is essentially based on the strategy of frequency tuning for controlling the powerful behavior of bat swarm and it is possible to manipulate the balance between exploration and exploitation by tuning the algorithm dependent parameters.

Therefore, the effective approach in this research analysis is to draw out the main element features useful in the classification of ECG beats. With this sub-section 5 key features from 300 samples are extracted using bat algorithm. Bats [18] will be the fascinating band of birds. They count on echolocation which has a total of 1200 varieties of bats [19]. Over fifty percent of these rely on echolocation to find their prey. Bats echolocation capacity usually, a number of the bats use a superior and sophisticated sense ability to hear. The noises are released that bounces echo back again from the pests or the things in their route. From these echoes, bats can identify the pests or thing lengths which are using their current position and also estimate how big are insects or things within a portion of the second [20].

The methods for extracting features generally give rise to a more features, where some feature might be insignificant. In this research analysis, therefore, the effective approach is to identify the main elements, which is useful in classifying the trusted IoT devices. A Bat algorithm is used to extract 5 significant features over 1000 samples.

The steps of bat algorithm isgiven below.

Step 1: Determine the fitness function for the random creation of bat inhabitants.

Step 2: Define pulse rate occurrence, parameters and factors of loudness, pulse and frequency (A, r and f).

Step 3: While $t<T_{max}$

Step 4: Frequency modification is carried out to obtain new solution

Step 5: Update the position and velocity

Step 6: if ($rand>\underline{r_i}$)

i. Decide optimal solution between obtained better solutions

ii. Generate a local solution from the optimal solution.

Step 7: end if

Step 8: Generate new solutions using random movement of bats.

Step 9: if (rand $<A_i$ and $f(x_i) <f(x)$)

i. Increase the pulse rate (r_i) value and lower the loudness value (A_i)

ii. Find new solution using loudness and pulse rate.

Step 10: End if

Step 11: Get bats ranking

Step 12: Discover the current new best solution.

Step 13: End iteration and display the obtained features.

3.3.1. Data Partitioning.

The features are labeled once the datasets of the extracted features are created and it is chronologically divided into three separate sets. After the selected features have been extracted, we can use it. The first set (DSs) is used to generate a number of classifiers of single-session. In the second set (DSm), the parameters of a multi-session classifier are optimized. The third set (DStest) is used to evaluate the proposed study and derives performance measures.

3.4. CapsNet

As stated earlier, the study develop a CapsNetarchitecture to accurately classify the trusted and untrusted IoT from the Dstest.

Capsules are neuron groups that represent various parameters of activity for such neurons, and the length of these vectors indicates that a specific entity is likely to exist. CNN weaknesses are mainly linked to the layers of pooling. As a result, these lays are substituted in the capsule networks by a more suitable criteria named' routing by agreement' based on this criteria, but their coupling coefficients are not the same in all parent capsules in the next layer. The output of parent capsules is predicted in each Capsule and the cup coefficient between these two capsules is increased, if this prediction conforms to the actual output of the parent capsule. In view of u_i as capsule output i, its prediction for parent capsule j is as calculated

$$u_{j|i} = W_{ij} u_i \tag{1}$$

whereu^j|i is the vector prediction of the jth capsule's output at a higher level, which is calculated by Capsule i in the below layer, and W_{ij} is the weighting matrix that must be apprised in the backward pass. The coupling coefficients c_{ij} are calculated by using the following softmax function based on the conformation level between the capsules in the below and the parent capsules.

$$c_{ij} = \frac{\exp\left(b_{ij}\right)}{\sum_k \exp\left(b_{ij}\right)} \tag{2}$$

where the log likelihood (b_{ij}) is that if the capsule i is linked with capsule j and at the beginning of the routing process it is set to 0. The input vector for the parent capsule j is therefore calculated accordingly.

$$s_j = \sum_i c_{ij} u_{j|i} \tag{3}$$

Finally, the following nonlinear squash function avoids exceeding one of the output vectors of Capsules and forms the end output of each Capsule based on its initial vector value defined in Eq. (3)

$$v_j = \frac{\left\|s_j\right\|^2}{1+\left\|s_j\right\|^2} \frac{s_j}{\left\|s_j\right\|} \tag{4}$$

where, $s_j$and$v_j$is regarded as the input and output vector of j. The log probabilities in the classification process should be updated based on the agreement between v_j and u_j, using the fact that if the two

vectors agree, a large internal product will be provided. Therefore agreement is calculated as follows in order to update log probabilities and coupling coefficients

$$a_{ij} = v_j \cdot u_{j|i} \tag{5}$$

Each capsule k in the last layer has a loss function l_k that puts a high percentage value on capsules with long output instantiation parameters when there is no entity. The loss function l_k is calculated accordingly

$$l_k = T_k \max(0, m^+ - \|v_k\|)\ 2 + \lambda(1-T_k)\ \max(0, \|v_k\| - m^-)^2 \tag{6}$$

T_k is 1 exist if k is present, or else T_k is 0. Before the learning process, the terms or hyper parameters m^+, m^- and λ indicated. In [9], the original network capsule architecture consists of one convolutional filter layer and two capsule layers. It also has three completely linked layers of neurons that attempt to re-build the input using the capsule instantiation parameters that are associated with the true label.

4. Results and discussions

We evaluate our method using the third dataset, Dstest in Contiki IoT simulator. The results indicate that by analyzing network traffic we can distinguish between Ips that belong to IoT devices and Ips that belong to PCs and smartphones. Smartphones were classified by analyzing the "user agent" HTTP property and PCs with single-session classifier. The proposed method is compared against Machine learning, filters and stepwise regression wrappers and tree-based clustering. The study uses five

The study is evaluated using accuracy, specificity, sensitivity, F-measure, execution time and mean error rate with following metrics

TP– True Positive, trusted IoT devices as trusted devices

TN– True Negative, trusted IoT devices as untrusted devices

FP – False Positive, untrusted IoT devices as untrusted devices

FN– False Negative,untrusted IoT devices as trusted devices

4.1. Accuracy

Accuracy is defined as follows

$$accuracy = \frac{TN + TP}{TP + TN + FN + FP} \tag{7}$$

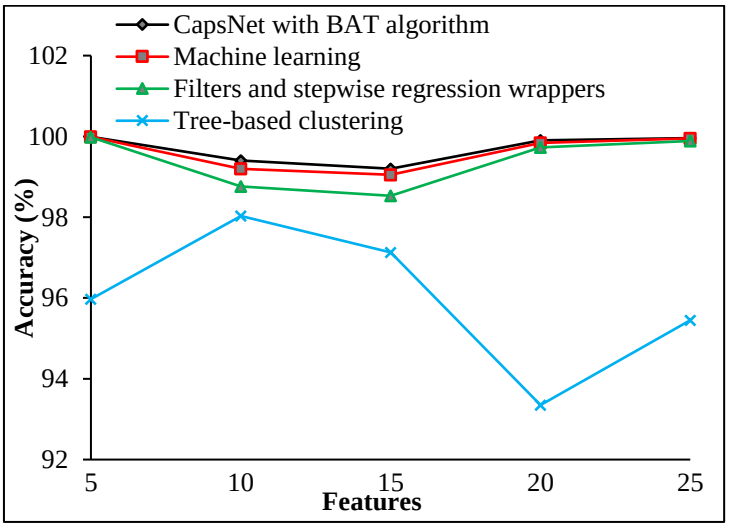

Figure 2. Accuracy

Table.1. Accuracy

Features	Tree-based clustering	Filters and stepwise regression wrappers	Machine learning	CapsNet with BAT algorithm
5	95.97	99.98	99.99	99.99
10	98.03	98.76	99.2	99.4
15	97.13	98.53	99.05	99.2
20	93.35	99.73	99.84	99.9
25	95.45	99.89	99.95	99.96

The Figure 2 and Table 1 shows the results of accuracy between different classifiers. The result shows that the proposed CapsNet with BAT algorithm achieves improved classification accuracy with increasing features, however, the presence of insignificant features reduces the accuracy rate. This can be evident with 10 and 15 features and on other hand, the significant features (say 5) produces crisp output.

4.2. Sensitivity

Sensitivity determines the ability of the test to find or correctly discover a trusted IoT unit. This is expressed mathematically as:

$$sensitivity = \frac{No.\ of\ \ TP}{No.\ of\ \ TP + No.\ of\ \ FN} \tag{8}$$

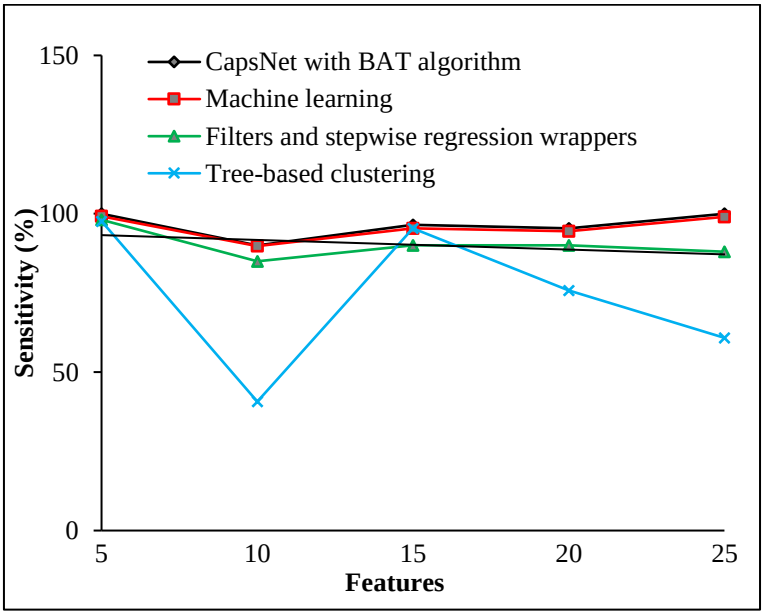

Figure 3. Sensitivity

Table.2. Sensitivity

Features	Tree-based clustering	Filters and stepwise regression	Machine learning	CapsNet with BAT algorithm

		wrappers		
5	97.67	98.1	99.2	100
10	40.71	85	89.8	90
15	95.38	90	95.4	96.48
20	75.84	90	94.5	95.38
25	60.85	88	99	100

The Figure 3 and Table 2 shows the results of sensitivity between different classifiers. The result shows that the proposed CapsNet with BAT algorithm achieves improved sensitivity rate than other methods.

4.3. Specificity

The specificity concerns the potential of the test to identify reliable devices without the condition of network appropriately. Consider an example of an IoT device or non-IoT device network test for diagnosis. The specific character of the testing results is the percentage of trusted devices not known to have a network condition. This can be mathematically written as:

$$specificity = \frac{No.\ of\ \ TN}{No.\ of\ \ TN + No.\ of\ \ FP} \tag{9}$$

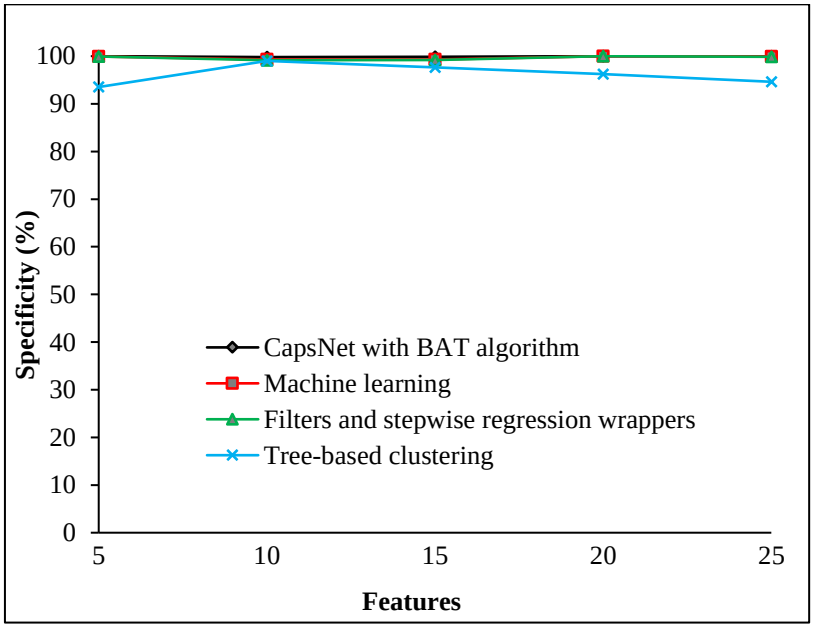

Figure 4. Specificity

Table.3. Specificity

Features	Tree-based clustering	Filters and stepwise regression wrappers	Machine learning	CapsNet with BAT algorithm
5	93.52	99.94	99.96	100
10	99	99.15	99.3	99.8
15	97.66	99.19	99.3	99.88
20	96.23	100	100	100
25	94.65	99.88	99.9	100

The Figure 4 and Table 3 shows the results of specificity between different classifiers. The result shows that the proposed CapsNet with BAT algorithm achieves improved specificity rate than other methods.

4.4. F-Measure

The F-measure is an accuracy test with *P* and *R* metrics. This F-measure therefore calculates the average value for PR. If it's 1 and 0, the average value for PR is predicted best. Strategy F1 is considered to be the harmonic mean of precision and reminder. It is assumed that an excellent classifier actually has a high measurement F1, which means that the classifier is good at both precision (*P*) and recall (*R*)..

$$F1 \text{ strategy} = (2PR)/(P + R) \quad (10)$$

P is called as the precision, where the number of correct positive outcomes divided by all positive outcomes.

R is defined the recall, where the number of positive correct results divided by the number of all positive results.

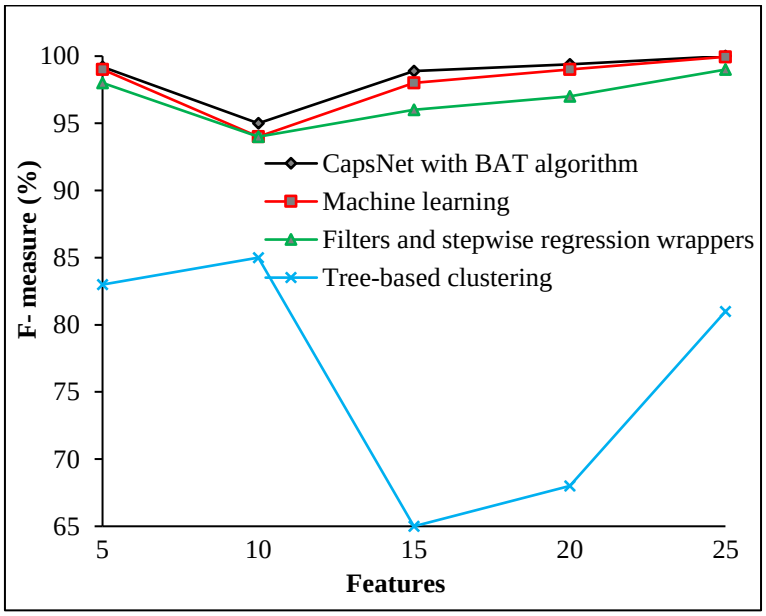

Figure 5. F-Measure

Table.4. F-measure

Features	Tree-based clustering	Filters and stepwise regression wrappers	Machine learning	CapsNet with BAT algorithm
5	83	98	99	99.2
10	85	94	94	95
15	65	96	98	98.9
20	68	97	99	99.4
25	81	99	99.95	100

The Figure 5 and Table 4 shows the results of F-measure between different classifiers. The result shows that the proposed CapsNet with BAT algorithm achieves improved F-measurethan other methods.

4.5. Execution Time

The run time shows that in comparison with the previous work, the proposed algorithm gives the result as early as possible.

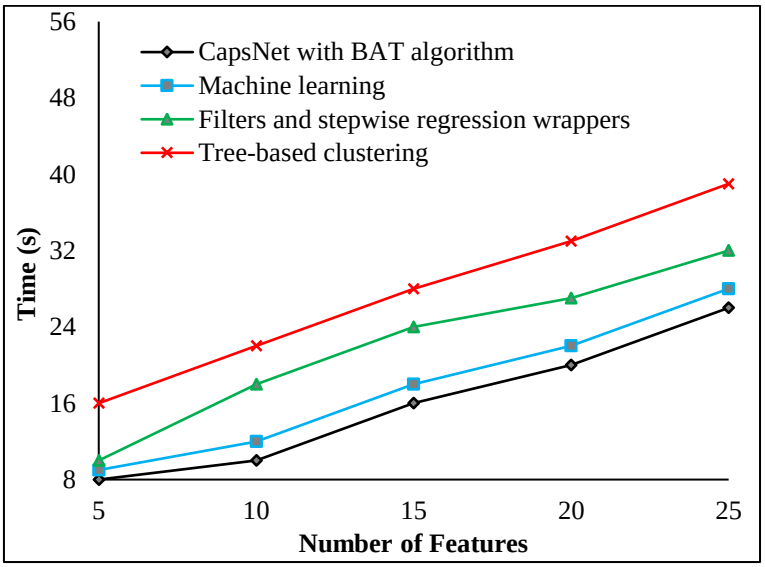

Figure 6. Execution Time

Table.5. Execution Time

Features	Tree-based clustering	Filters and stepwise regression wrappers	Machine learning	CapsNet with BAT algorithm
5	16	10	9	8
10	22	18	12	10
15	28	24	18	16
20	33	27	22	20
25	39	32	28	26

The Figure 6 and Table 5 shows the results of Execution Timebetween different classifiers. The result shows that the proposed CapsNet with BAT algorithm achieves reduced execution timethan other methods.

4.6. MAE

The mean absolute error MAE is defined as the time required to evaluate how the true value is predicted and how the final result are revealed. It is given by Eq.(11)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i| \tag{11}$$

The MAE is also defined as the average value of the absolute errors, which obtains the true and predicted value.

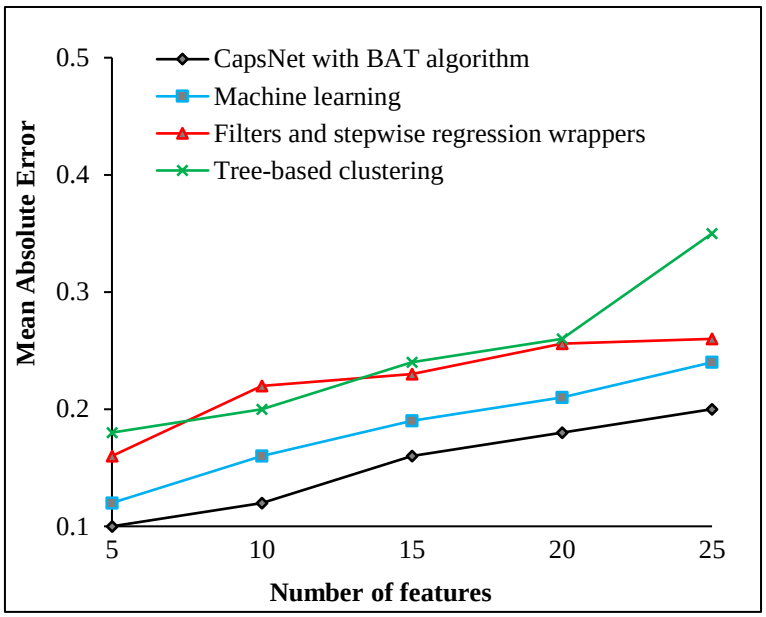

Figure 7. Mean Absolute error

Table.6. Mean Absolute error

Features	Tree-based clustering	Filters and stepwise regression wrappers	Machine learning	CapsNet with BAT algorithm
5	0.18	0.16	0.12	0.1
10	0.2	0.22	0.16	0.12
15	0.24	0.23	0.19	0.16
20	0.26	0.256	0.21	0.18
25	0.35	0.26	0.24	0.2

The Figure 7 and Table 6 shows the results of Mean Absolute errorbetween different classifiers. The result shows that the proposed CapsNet with BAT algorithm achieves reduced Mean Absolute errorthan other methods.

5. CONCLUSIONS

In this paper, we classified the trusted and non-trusted IoT devices in a network using CapsNet classification. This method uses a series of operations to improve the rate of classification of trusted by monitoring effectively the network traffic.CapsNetiterativelyimprovestheclassification of IoT devices and accurately identifies the trusted IoT devices based on labelled data. The simulation result shows that the proposed deep learning architecture for IoT device classification performs accurate identification of trusted and non-trusted IoT devices on the labeled network traffic data. The present system achieves an accuracy rate of 99.4%, which is higher than other methods. This validates that the use of meta-heuristic learning using Bat algorithm for feature extraction and deep learning CapsNet for classification improves effectively the rate of trusted IoT device classification. Further, it is believed that the proposed method recognizes accurately and automatically the unauthorized IoT devices connections in a computer network of an enterprise. The classification of untrusted connection helps to mitigate the violations associated with the enterprise operational policies. In future, the study tends to use unlabeled data from the extracted features for the classification of IoT devices in a large scale network.

REFERENCES

[1] Miorandi, D., Sicari, S., De Pellegrini, F., &Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad hoc networks*, *10*(7), 1497-1516.

[2] Singhal, A., &Ou, X. (2017). Security risk analysis of enterprise networks using probabilistic attack graphs. In *Network Security Metrics* (pp. 53-73). Springer, Cham.

[3] Lakhina, A., Crovella, M., &Diot, C. (2005, August). Mining anomalies using traffic feature distributions. In *ACM SIGCOMM computer communication review* (Vol. 35, No. 4, pp. 217-228). ACM.

[4] Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, E. D., & Taft, N. (2004, June). Structural analysis of network traffic flows. In *ACM SIGMETRICS Performance evaluation review* (Vol. 32, No. 1, pp. 61-72). ACM.

[5] Iglesias, F., &Zseby, T. (2015). Analysis of network traffic features for anomaly detection. *Machine Learning*, *101*(1-3), 59-84.

[6] Afshar, P., Mohammadi, A., &Plataniotis, K. N. (2018, October). Brain tumor type classification via capsule networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 3129-3133). IEEE.

[7] Zhao, W., Ye, J., Yang, M., Lei, Z., Zhang, S., & Zhao, Z. (2018). Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*.

[8] Shahroudnejad, A., Afshar, P., Plataniotis, K. N., &Mohammadi, A. (2018, November). Improved explainability of capsule networks: Relevance path by agreement. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (pp. 549-553). IEEE.

[9] Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems* (pp. 3856-3866).

[10] Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J. D., Ochoa, M., Tippenhauer, N. O., &Elovici, Y. (2017, April). ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis. In *Proceedings of the symposium on applied computing* (pp. 506-509). ACM.

[11] Iglesias, F., &Zseby, T. (2015). Analysis of network traffic features for anomaly detection. *Machine Learning*, *101*(1-3), 59-84.

[12] Bhuyan, M. H., Bhattacharyya, D. K., &Kalita, J. K. (2016). A multi-step outlier-based anomaly detection approach to network-wide traffic. *Information Sciences*, *348*, 243-271.

[13] Sudharson D, Prabha D, "A novel machine learning approach for software reliability growth modelling with pareto distribution function", Soft Computing, Springer Publisher, May 2019.

[14] V.R. Azhaguramyaa, K. Srinivasan, P. Rajasekar, A. Lokeshwaran and T.L. Manoj Kumar, "A Study of Specimen Classification of an Image Using Machine Learning ", Journal of Advanced Research in Dynamical and Control Systems, vol. 10(12), pp. 1037-1039, 2018.

[15] A.Pushpalatha, "Survey on Swarm Optimization Algorithms," Journal of Advanced Research in Dynamical and Control Systems, vol. 12, special issue, pp. 1117-1120, 2018.

[16] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74). Springer, Berlin, Heidelberg.

[17] Yang, X. S., &HosseinGandomi, A. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, *29*(5), 464-483.

[18] Xin-She Yang and Xingshi He, "Bat Algorithm: Literature Review and Applications", *International Journal of Bio-Inspired Computation*, Vol. 5, No. 3, pp.141-149, 2013.

[19] IztokFister Jr et. al., "Particle Swarm Optimization for Automatic Creation of Complex Graphic Characters", *Chaos, Solitons and Fractals*, Vol. 73, pp. 29-35, 2015.

[20] Xin-She Yang, "Bat Algorithm for Multi-Objective Optimisation", *International Journal of Bio-Inspired Computation*, Vol. 3, No. 5, pp. 267-274, 2011.