Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 01
Pages: 20-30
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.1.003

Real Time Anomaly Detection Techniques Using PySpark Frame Work

Dr. G. Ranganathan

Professor,Department of Electronics & Communication Engineering,

Gnanamani College of Technology, NH-7, A.K.Samuthiram, Pachal P.O,

Namakkal, Tamil Nadu, India -637018.

Email id: profranganathang@gmail.com

Abstract: The identification of anomaly in a network is a process of observing keenly the minute behavioral changes from the usual pattern followed. These are often referred with different names malware, exceptions, and anomaly or as outlier according to the dominion of the application. Though many works have emerged for the detection of the outlier, the identification of the abnormality in the multiple source data stream structure is still under research. To identify the abnormalities in the cloud data center that is encompassed with the multiple-source VMWare, by observing the behavioral changes in the load of the CPU, utilization of the memory etc. consistently the paper has developed a real time identification process. The procedure followed utilizes the PySpark to compute the batches of data and make predictions, with minimized delay. Further a flat-increment based clustering is used to frame the normal attributes in the PySpark Structure. The latencies in computing the tuple while clustering and predicting, was compared for PySpark, Storm and other dispersed structure that were used in processing the batches of data and was experimentally found that the processing time of tuple in a PySpark was much lesser compared to the other methods.

Keywords: VMware, Cloud Data Center, Real Time Abnormality Detection, Pyspark Frame Work, Clustering (Flat-incremental)

1. Introduction

The anomalies cause deviation from the actual expected behavior of the system. They indicate the outliers such as the dangerous attacks, burdens and the glitches or the errors that are caused in the system. These are commonly referred with various names based on the dominion of the applications such as outliers, intrusions, attacks or even as failures and errors. As there are enormous amount of data produced in real time constantly, observing them keenly to detect changes caused in their normal behavior, utilizing the traditional system is insubstantial. Furthermore observing and correcting an abnormality in real time is even more complicated as it has to keep note of several characteristics and process the data's that are with

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 01
Pages: 20-30
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.1.003

complex nature. So devising an automatic detector to identify the abnormality and deciding the corrective measure in real time becomes essential, as more such methods devised to detect the anomaly, the paper also tries to develop a real time anomaly detection procedure using the Pyspark to observe the data streams in the cloud data center with the multiple-Source VMware.

The data centers in the cloud are private clouds that provide infrastructure as service, these usually meant to provide a safe and secure service provisioning based on the request made. The resources in the data center are managed according to the demands of the users and the availability of the resources. The infrastructures are either scaled up or down based on the demands of the clients. To provide a satisfactory service the data center has to maintain a real time observer to note down constantly the functioning to observe the requirements as well as the changes in the behavior, it also performs a rescheduling to balance the system. For instance supposes a client is need of more resources in the midst of a computation, the observers immediately monitor them and demands and allots substantial resources instantly.

Computing the streams of data in online is a delay sensitive process, so it becomes essential to process the enormous data flow at high speed and transform it into valuable information's, that are to be furthermore processed without complications at lower latencies even when the data flow at a higher speed. To construct such a scalable system with the capacity of operating in real time. The paper develops a multiple source data stream observer that utilizes the PySpark, to detect the outliers, the devised procedure uses the Kafka as a message broker to deliver the messages in proper order as stored by the user and the flat incremental clustering is done to develop a frame the normal activities based on the normal dataset provided for training. The clusters are produced dynamically and are not fixed.

Some of the essential blocks and the process for constructing the real time anomaly observer are the discussed below

i. Cloud Data Center: It is storage house that utilizes the appliances of the cloud storage as its infrastructure, and also delivers the essential computer components that are demanded by the users. It handles the infrastructure and the storage system with the Microsoft, VMware, and Open stack using a software to handle the resources. The method proposed in the paper uses a datacenter with the VMware.

ii. Scheduling: The method follows a machine learning based scheduling, to examine the data streams and applies clustering that is flat incremental. However the classification finally is done applying the outlier identification strategy in case of unsupervised learning and in case of the supervised learning uses the

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 01
Pages: 20-30
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.1.003

classification model that is trained based on the normal and the abnormal activities and used in predicting the anomaly.

iii. VMware: The commands such as the TOP, and ESXTOP delivers the “performance data” of the virtual machines that are captured periodically. These information’s provide with the detail of the CPU and he memory Usage and reschedules the resource allocation dynamically if the usage cross the limitation set.

iv.PySpark: It is a Python based API developed using python to aid the Apache Spark, The Appache Spark written in the Scala is incorporated with the Python, Java, SQL, Scala and R languages , the spark is capable of processing enormous amount of Big Data in parallel, The library Py4j is used to work with the PySpark. The PySpark is utilized as it is more beneficial because of the easy integration offered with other languages, “Resilient distributed dataset”, “greater speed in processing” and “caching and disk persistence”

The PySpark programming includes “Resilient distributed dataset” that are fault tolerant ad distributed in nature, dual operations the transformation and the actions are performed over the data set where the transformation is done to on input data to transform and set them and action is performed by the python. Python performs the filtering and the sorting out process on the data collection that are well structured or partially structured and arranged in columns that are named. The machine learning algorithms are used to transform the input data’s, modify it using the transform algorithm and generate a trained model using the estimator algorithm and the Fit_function. The main reason why the PySpark is preferred is, it provides an easy interface, easy to learn syntax and languages, better readability, maintenance and familiarity, simple, easy and comprehensive interface and highly preferred for implementing machine learning algorithms. The proposed framework is assigned with related works in 2. The prosed work in 3. Results analysis in 4 and Conclusion in 5.

2. Related Works

Charikar, et al [1] "Incremental clustering and dynamic information retrieval." Chandola, et al [2] elaborates the “A survey on Anomaly detection." Zaharia, et al [4] presents “Fast and interactive analytics over Hadoop data with Spark." Kumari, R., et al [5] discusses the “Anomaly detection in network traffic using K-mean clustering." Praveena, A. et al [6] performs he “Anonymization in social networks as a survey on the issues of data privacy in social network sites."

Pwint "Network et al [7] conducts the “Traffic Anomaly Detection based on Apache Spark." Tagliafico, et al [8] does the “Real time anomaly detection in network traffic time series."Choi, et al [9] performs the

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 01
Pages: 20-30
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.1.003

"Development of Scalable On-Line Anomaly Detection System for Autonomous and Adaptive Manufacturing Processes."

Smys, S.et al [10] presents the “DDOS Attack Detection In Telecommunication Network Using Machine Learning." Duraipandian, M. et al [11] conducts the “Performance Evaluation of Routing Algorithm for Manet Based on the Machine Learning Techniques." Jyothirmai, Pondi, et al [12] elaborates the “Secured self-organizing network architecture in wireless personal networks."

3. Proposed Work

The Proposed work is about an anomaly observer who could deliver its service s in real time. It is encompassed with the blocks to perform, “pre-processing, training, predicting, scheduling, updating and rescheduling” as shown in the flow diagram displayed below in figure.1

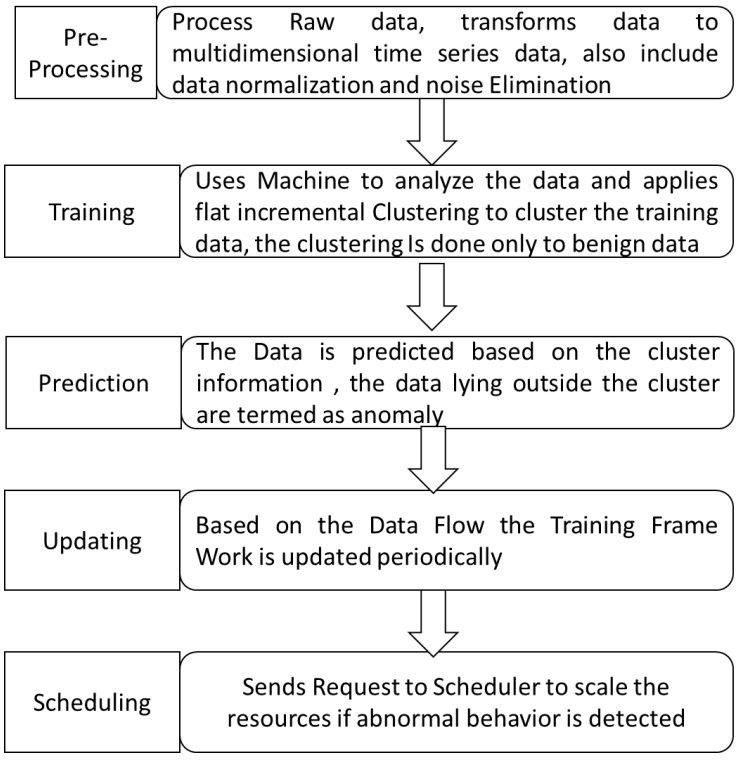

Figure. 1 Proposed Flow Diagram

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 01
Pages: 20-30
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.1.003

The fundamental process in the constructing the anomaly observer with the capability to observe the abnormality in the real time is depicted in the figure.1 the incoming data are pre-processed and data fixed for training is used to form clusters, as explained above the clusters are frame only for the data that appears to be normal and that which lays outside the clusters are coined as abnormal activity. Once the construction is completed the execution of the requisitions begin in the virtual machines, according to the size of the tasks the metrics of the CPU change and unfits the clusters , so this would be treated abnormal in the prediction stage, to avoid this the training model is scaled up periodically accepting the alterations. Any deviation in the regular activities after the completion of the prediction is termed as anomaly and notified to the scheduler of the resource. The algorithm below in figure.2 explains the process of flat incremental clustering.

```
procedure Flat Incremental
Clustering
 TC = Total Cluster
 {
 k = number. Of. cluster
 TC= 0
 }
for each data point n do
 {
 Fit  = Fits In Any Cluster(n)
 if  Fit == false then
 Create Cluster()
 total cluster  =  TC+ 1.
 }
if  TC> k then
 {
Merge Two nearest Cluster To One ()
 }
else
Update Cluster Info()
```

Figure. 2 Algorithm for Flat Incremental Clustering

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 01
Pages: 20-30
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.1.003

3.1. PySpark Based Anomaly Detection

The Cluster of the VMware is comprised of five VMware hyper servers with each server equipped with "Intel (R) Xeon (R) CPU E5-2695 v2 2.40 GHz processor, 64 GB DDR3 RAM, a 4TB hard disk and a dual NIC Card." Every processor is provided with two sockets with the capacity to afford twelve cores, every VM is set up with eight V CPU, sixteen Gigabytes of DDR3 and one terabyte hard disk. The resources are shared by the VM to provide a better performance. Apart from the aforementioned the VM is implemented with the "Linux Centos V6.5 64 bit operating system with Java JDK/JRE v.17." the outlier identifier was designed using the PySpark and the messages disposal in proper order was taken care by Kafka

The detection process is performed in two strides training and testing, the PySpark constructs the training model using the two algorithm transform and Action, the transform receives the input data that is the tuple/ or the data streams, extracts values and identifies a closest cluster for it, if a cluster is not found the data extracted is termed as the new centroid of the cluster and frames another cluster otherwise updates the existing cluster, later the "cluster is picked form its listing and updated by acquiring the weighted average of the list of the data instance." The developed clusters are kept in memory due to the state-full property of the PySpark, the cluster alone is stored explicitly, and this is termed as the action of PySpark. The similarity (S) for merging a two closest clusters are measured according to its distance (D) as shown in equation 1

$$S = 1/(D + 1) \quad (1)$$

Based on the equation closer the distance more will be the similarity and vice versa. Once the training is completed the testing is performed, if the data doesn't fit into any of the clusters it is termed anomaly, and if the distance of the cluster center point and the data are less than the radius of the cluster the data are considered as benign and added to the clusters else termed as anomaly. The accuracy of the proposed frame work is maintained by periodically scaling up the clusters, the complete set up is retained in the HDFS for standby.

4. Results Analysis

The data set were gathered in real time by subjecting the virtual machine to multiple of tasks, and capturing the CPU's performance metrics. The data gathered were framed as the benign data and further metrics of the CPU's were programmatically increased to produce the abnormal data. So the evaluation process was

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 01
Pages: 20-30
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.1.003

carried out using two sets of data one normal and the other one abnormal. The table.1 below shows the configuration of the PySpark Cluster.

Components	Values
Components working in parallel for Data Generation	5
Number of predictions	8
Number of data points	Dataset 1= 12,000 Dataset 2 =10,000 Dataset 3 =11,000
Number of clusters	Dataset 1= 63 Dataset 2 =79 Dataset 3 =85
Number of iterations in training	10,000

Table.1 Components Used

The accuracy of the proposed model in terms of "True Positive Rate", True Negative Rate, False Positive Rate and False Negative Rate" is depicted in the figure. 3 based on the data set collected on three different scenarios.

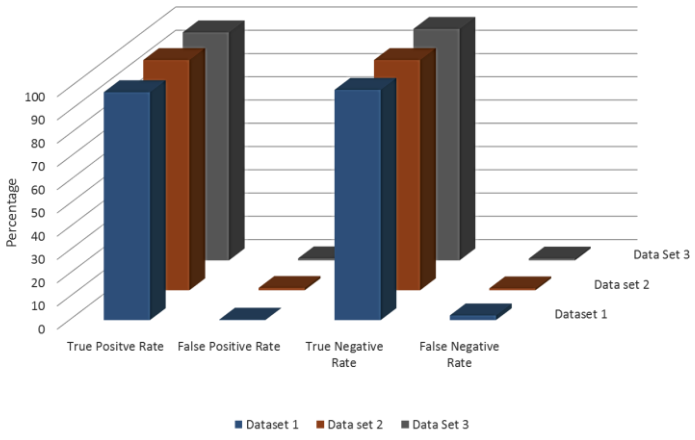

Figure.2 "True Positive Rate", True Negative Rate, False Positive Rate and False Negative Rate"

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 01
Pages: 20-30
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.1.003

The results of the developed frame work is compared with the Strom and the Spark programmed in Scala on the terms of the processing latencies, latency observed in examining the data , the time consumed in processing the entire data and the level of accuracy achieved.

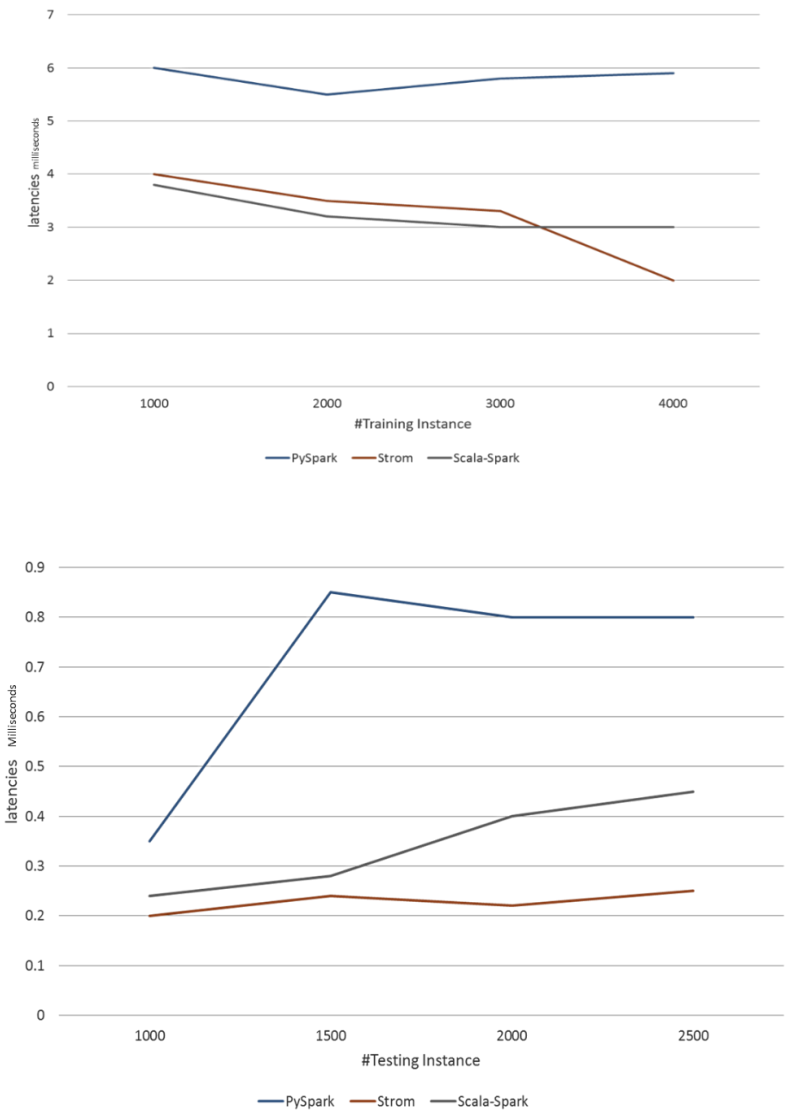

Figure.3 Processing Latencies Training and Testing

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 01
Pages: 20-30
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.1.003

The figure.3 shows the comparison between the PySpark, Strom and the Scala-Spark for testing and as well as training , the results observed shows that the proposed frame has much lesser processing and analyzing latencies than the other two methods.

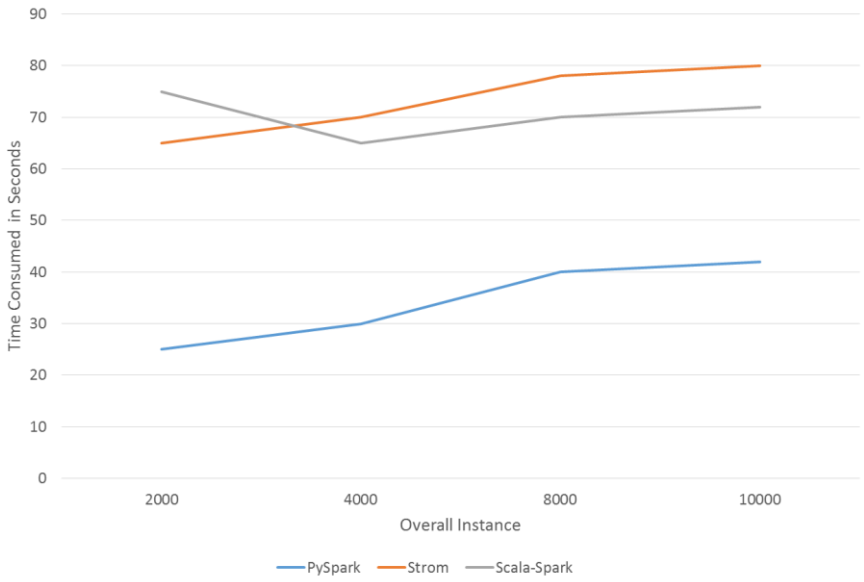

Figure.4 Time Consumed

The figure.4 depicts the total time consumed in processing the entire data by the PySpark and its comparison with the other methods such as the Strom and the Scala-Spark. The figure.5 further depicts the overall accuracy percentage achieved but eh each model in identifying he anomaly.

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 01
Pages: 20-30
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.1.003

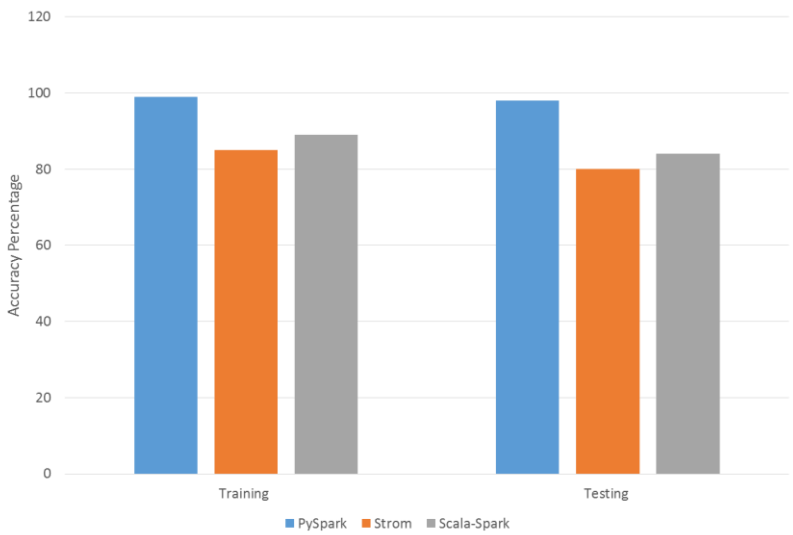

Figure.5 Average Accuracy Rate

5. Conclusion

The paper has achieved the implementation of an anomaly detector to conduct its observation in real time using the PySpark. The capability of the frame work in handling the data, its fault tolerance, its flexibility, and the guaranteed delivery were much better compared to the existing methods. The results observed in evaluating the proposed frame work was much convincing in terms of the processing latencies, time consumption and accuracy compared to the Strom and the Scala-Spark. In future the paper scopes to improve the model including the algorithms of machine learning and observing the performance changes and associate them in identifying the abnormalities.

References

[1] Charikar, Moses, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. "Incremental clustering and dynamic information retrieval." *SIAM Journal on Computing* 33, no. 6 (2004): 1417-1440.

[2] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41, no. 3 (2009): 1-58.

ISSN: 2582-2012 (online)

[3] https://kafka.apache.org/intro

[4] Zaharia, Matei, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy Mccauley, M. Franklin, Scott Shenker, and Ion Stoica. "Fast and interactive analytics over Hadoop data with Spark." *Usenix Login* 37, no. 4 (2012): 45-51.

[5] Kumari, R., M. K. Singh, R. Jha, and N. K. Singh. "Anomaly detection in network traffic using K-mean clustering." In *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pp. 387-393. IEEE, 2016.

[6] Praveena, A., and S. Smys. "Anonymization in social networks: a survey on the issues of data privacy in social network sites." *Journal of International Journal Of Engineering And Computer Science* 5, no. 3 (2016): 15912-15918.

[7] Pwint, Phyo Htet, and Thanda Shwe. "Network Traffic Anomaly Detection based on Apache Spark." In *2019 International Conference on Advanced Information Technologies (ICAIT)*, pp. 222-226. IEEE, 2019.

[8] Tagliafico, Sergio Martınez, Gastón Garcıa González, Alicia Fernández, Gabriel Gómez Sena, and José Acuna. "Real time anomaly detection in network traffic time series."

[9] Choi, Seunghyun, Sekyoung Youm, and Yong-Shin Kang. "Development of Scalable On-Line Anomaly Detection System for Autonomous and Adaptive Manufacturing Processes." *Applied Sciences* 9, no. 21 (2019): 4502.

[10] Smys, S. "Ddos Attack Detection In Telecommunication Network Using Machine Learning." *Journal of Ubiquitous Computing and Communication Technologies (UCCT)* 1, no. 01 (2019): 33-44.

[11] Duraipandian, M. "Performance Evaluation Of Routing Algorithm For Manet Based On The Machine Learning Techniques." *Journal of trends in Computer Science and Smart technology (TCSST)* 1, no. 01 (2019): 25-38.

[12] Jyothirmai, Pondi, Jennifer S. Raj, and S. Smys. "Secured self-organizing network architecture in wireless personal networks." *Wireless Personal Communications* 96, no. 4 (2017): 5603-5620.

ISSN: 2582-2012 (online)