KSCST

ಕರ್ನಾಟಕ ಸರ್ಕಾರ

Department of Science and Technology
Government of India

KARNATAKA STATE COUNCIL FOR SCIENCE AND TECHNOLOGY

Indian Institute of Science campus, Bengaluru

Telephone: 080 -23600978, 23341652 || Email: spp@kscst.org.in
Website: www.kscst.iisc.ernet.in/spp.html or www.kscst.org.in/spp.html

46th SERIES OF STUDENT PROJECT PROGRAMME

SYNOPSIS

(46S_BE_3815)

1.	**Title of the Project: Adaptive Anomaly detection system for Software Defined Networks**
2.	**Name of the College: RV College of Engineering**
3.	**Department: Computer Science and Engineering**
4.	**Name(s) of Project guide(s):** 1. **Name: Prof. SNEHA M** **Email id: sneham@rvce.edu.in** **Contact No.: 8792834722** 2. **Name: Dr. ASHOK KUMAR A R** **Email id: ashokkumarar@rvce.edu.in** **Contact No.:8497042779**
5.	**Name of the Student(s):** **Name: KEERTHAN KUMAR A** **Email id: keerthankumara.cs19@rvce.edu.in** **Mobile No: 9110824698** **Name: OMPRAKASH** **Email id: omprakash.cs19@rvce.edu.in** **Mobile No: 8884748321** **Name: MOHAMMED ZAID SIKANDER** **Email id: mohammedzaids.cs19@rvce.edu.in** **Mobile No: 9945493556**
6.	**Keywords:** Software Defined Networks, Ryu Controller, Mininet, Machine Learning, MongoDB
7.	**Introduction/Background:**

	Software Defined Networking has changed the way switches work, and industries have started adopting it. The work leverages the architectural advantages of SDN to detect count based attacks like DoS, in real-time, by utilizing machine learning models like Restricted Boltzmann Machine (RBM), Variational Auto Encoder (VAE), and Neural Basis Expansion Analysis for Time Series (N-BEATS). Traditional networks had to be monitored on every critical point, for such attacks, but with SDN only the centralized controller needs to be used to detect such attacks. The existing works rely on some networking dataset, or use a simulated one. Using simulated data guarantees the working for changing network scenarios. Some have employed a dynamic threshold, since the network statistics keep changing over time. But most of these fail to give an actual real-time detection implementation, which is what this project aims to accomplish. Also none of the works have used any sort of hardware to generate data. By utilizing hardware in one of the modules, realistic data is generated, that mimics the actual network. This ensures better results and prediction. By utilizing a sliding window technique, dynamic thresholding through Z-score and machine learning, the system can provide attack detection within the SDN architecture.
8.	**Objectives:** The system should be able to detect DoS like attacks in real-time (less the one second) by analyzing traffic patterns of the incoming data. Here are the objectives laid out: • Developing attack detection system based on SDN architecture • Implementing a robust solution that can analyze network traffic in real-time • Utilize machine learning to accurately distinguish anomalies from normal data • Generation of dataset from network emulation and using that to train the model • Provide a user-friendly frontend interface for visualizing network traffic patterns • Optimize system performance to handle large-scale networks • Enable adaptability to changing network environments • Facilitate control of the system from a single point
9.	**Methodology:** The architecture of the system is shown in Figure 1. Blue boxes indicate inputs to the system. There is a topology setup on the network emulation tool, which consists of a single switch and multiple nodes connected to it. Nodes transfer data to other nodes through the use of some tools. Majorly three kinds of traffic were focused on: video streaming traffic, file transfer traffic and http web traffic. Having multiple kinds of traffic ensures the realistic nature of the data and helps the models to adapt to real-life adaptations easily.

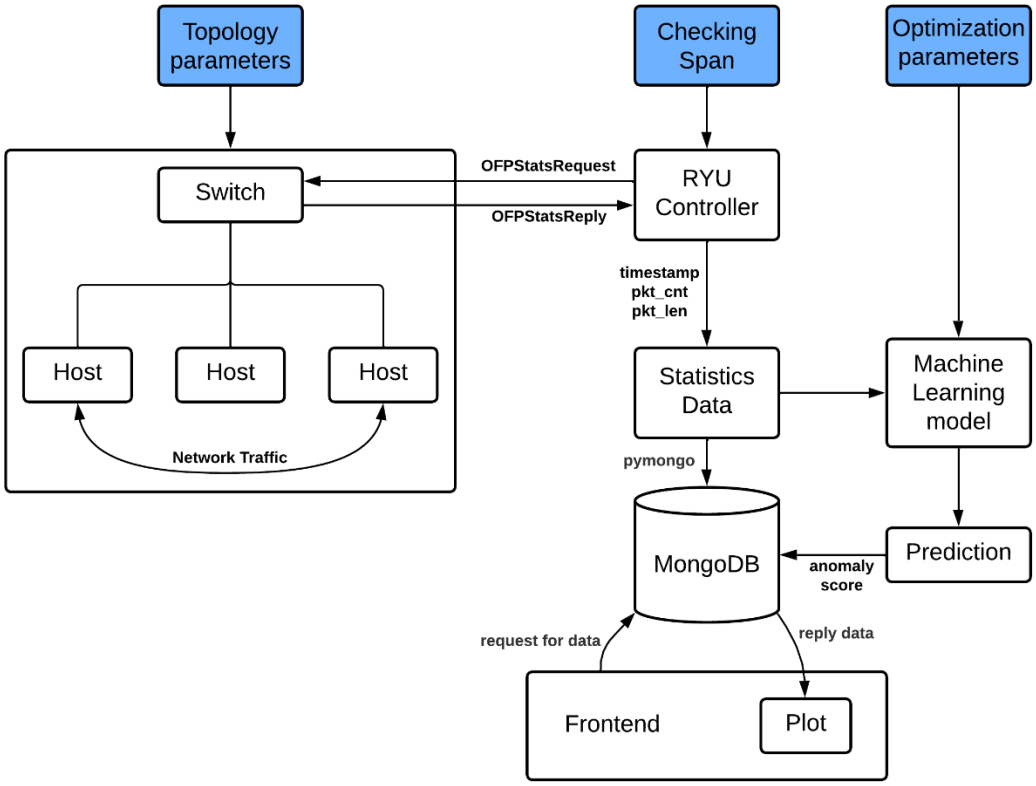

Figure 1. System Architecture

The system utilizes an approach where the models are trained on Google colab, and the trained model is transferred to the controller for prediction. This ensures that controller's resources are utilized efficiently, leaving more resources for the important work. The SDN controller periodically pings the switches for getting packet statistics. From this, the packet count is taken and used to detect the anomaly. The data generated is fed to a database, which will be further used to plot the data on the frontend, and also fed to a machine learning module, which gives the anomaly score. This anomaly score is normalized and compared with its usual mean and standard deviation to fix the threshold.

The system compared the network emulation tool on software (emulation on virtual machine), as well as on hardware (emulation on raspberry pi zero). By implementing on hardware, the realistic nature of traffic was revealed, and the models were able to recognize the probability distributions and give accurate predictions. The three models were also compared with one another, which will be discussed in the further section.

10. **Results and Conclusions:**

The models were compared with one another, to quantify the accuracy and predictions. Figure 2 shows the prediction times, where N-BEATS beats the other two models by predicting in just 25 milliseconds. This is extremely good for the real-time detection objective. And also we need to measure how well the model is able to detect an abnormal point from a normal point. This is shown in Table 1. Last column, the deviation score is calculated based on the difference of normal anomaly scores and abnormal scores, then

normalized to the normal scores. By looking at the numbers, a higher number denotes a better separation, and N-BEATS gave an exponentially better value than the other models.

The development of anomaly detection system has provided an effective solution for enhancing network security and detecting count based attacks. By leveraging the machine learning models, the system is able to accurately identify anomalies and distinguish them from the normal network behavior.

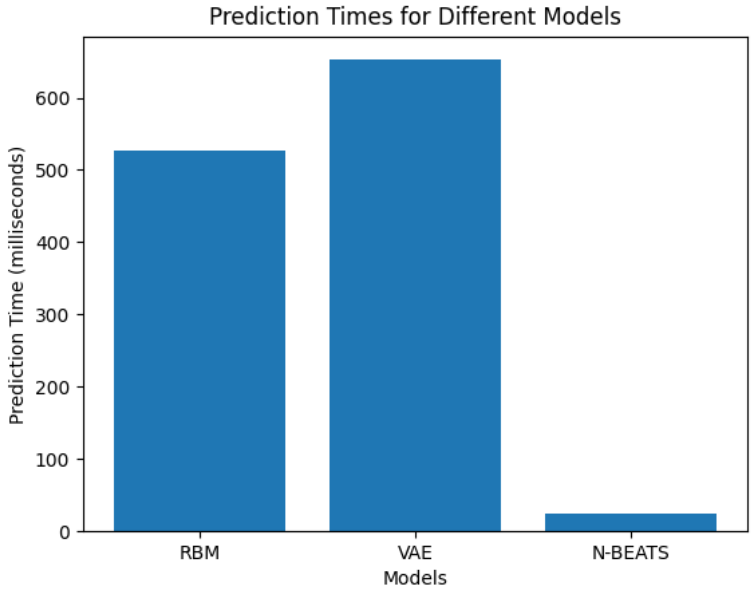

Figure 2. Model prediction times

Table 1. Anomaly score analysis for the models

Model	Mean of Normal scores	Mean of Abnormal scores	Difference	Deviation Score
RBM	0.328	0.810	0.482	1.469
VAE	0.036	0.631	0.595	16.527
N-BEATS	0.005	0.672	0.667	133.4

11. **Innovation of the Project:**

One notable innovation lies in the utilization of hardware for data generation, which generate a more realistic data. And the project's emphasis on real-time implementation sets it apart from other works in the domain, as many works do not give an actual real-time implementation.

12. **Scope for Future work:**

Although the developed anomaly detection system has achieved its objectives, there are other aspects it could aim. These are in the prospects of optimization. The system could better optimize the prediction by further fine-tuning the models and training on more data points. Also by extending the hardware implementation, an ideal dataset could be generated and could be

	published as a public dataset; as there is only one dataset that is there in this domain. The dataset available now (InSDN) has a limitation that it was generated entirely using virtualization. The authors also mention the same, and also state that better results could be achieved by using a hardware generated data. This can be aimed to accomplish in the future work extending this project by investing time and using more physical resources.