Contents

[Anomaly Detection and Trust Authority in Artificial Intelligence and Cloud Computing (Kashif Naseer Qureshi , Gwanggil Jeon , Francesco Piccialli) 2](#_Toc58939792)

[SDN based Anomaly Detection System 2](#_Toc58939793)

[1.Discovering 4](#_Toc58939794)

[1.1Network Topology Detector (NTD) 4](#_Toc58939795)

[1.2 Flow Information Detector (FID) 7](#_Toc58939796)

[2. Detection 7](#_Toc58939797)

[2.1 Malicious Traffic Detector (MTD) 7](#_Toc58939798)

[2.2 Isolation Module 10](#_Toc58939799)

[2.3 Storage Module 10](#_Toc58939800)

[Predicting Network Attack Patterns in SDN using Machine Learning Approach (Saurav Nanda , Faheem Zafari , Casimer DeCusatis , Eric Wedaa and Baijian Yang) 11](#_Toc58939801)

[Machine Learning Algorithms 11](#_Toc58939802)

[Algorithm 1 Machine Learning based predictor for SDN network attacks 12](#_Toc58939803)

[Effect of the Dataset 13](#_Toc58939804)

[Hybrid Deep Learning-based Anomaly Detection Scheme for Suspicious Flow Detection in SDN: A Social Multimedia Perspective (Sahil Garg, Member, IEEE, Kuljeet Kaur, Member, IEEE, Neeraj Kumar, Senior Member, IEEE, and Joel J. P. C. Rodrigues, Senior Member, IEEE) 14](#_Toc58939805)

[**Algorithm 1** Working methodology of the proposed scheme 15](#_Toc58939806)

[Dimensionality Reduction: Restricted Boltzmann Machine 15](#_Toc58939807)

[Classification: Support Vector Machines 16](#_Toc58939808)

# Anomaly Detection and Trust Authority in Artificial Intelligence and Cloud Computing (Kashif Naseer Qureshi, Gwanggil Jeon, Francesco Piccialli)

Table

Description automatically generated

## SDN based Anomaly Detection System

An SDN based network for IoT has various features that might affect the security and vulnerabilities such as centralized control, switch management, network services management, virtualized network, and monitoring unit. These features or services are an attractive point for attackers where they compromise business applications, controlling functions, and monitoring systems. The isolation of data and control planes between the controller and switches are realized by the well-defined programming interface. The flow tables in SDN plays a significant role. OpenFlow switch has a data path for that it is calculating by the controller. The flow table contains a predefined set of rules for network flow. The controller generates the flow rule and installs it in the flow table. The flow rules are based on three parameters including matching pattern, action related to matching pattern and statistics. The statistics contain priority, idle timeout, and hard timeout. Overflow attack targets the flow tables of forwarding devices in the data plane. This attack uses for packet dropping and cause of degradation of network performance. When the flow table is compromised it has serious malfunctioning on the flow tables. These flow rules are created and set by the controller by using proactive and reactive methods.

The proposed SDN-ADS based network is divided into three layers including management plane, data, and control plane. The SDN based system provides malicious detection for switches. The system calculates the forwarding path in the data plane and detects the attack paths. The SDN-ADS system is divided into three main modules including **Discovering, Detection, and Storage.** Then these three modules further have sub-components including for Discovering, **Network Topology Detector (NTD), Flow Information Detector (FID)**, for Detection, **Malicious Traffic Detector (MTD)** and Isolation and Storage.

Figure shows the proposed SDN-ADS system architecture.

Diagram

Description automatically generated

## 1.Discovering

In this module, the system discovers the topology and select the path. This module's basic duty is discovering the topology, setup the backup paths and then select the best path for packet forwarding. This module is further divided into the following sub-modules.

### 1.1Network Topology Detector (NTD)

In this module, the network topology discovers where active and backup paths determine by using the system. After this process, the controller sends data packets on the best path from source to destination and detects the switches that drop the packets and swap packets to reach the destination. In this step, the controller discovers the topology and then sends the packets to computed paths.

The process initiates to check priority queue to achieve the goal with less computation time and set the parameters for predecessor nodes and distance towards the switches to the controller. The optimal path and selected nodes are null until the path selection. The controller sets egress rules and makes a priority queue. Whenever, the discovery starts, the controller broadcasts the Discovery Message (DM) for the link layer to its connected switch. The Packet Out (P-O) message send to switches and then switches respond with Packet-In (P-I) message to the controller.

The same message sends to the predecessor and the same path is used for a reply with P-I and PO messages. In the proposed solution, if node n directly connected to controller C and queue is empty then set this node predecessor of the controller. After the discovery phase where priority queue is not empty, then retrieve the node with minimum cost. For every neighboring node, if receive DM packet and the distance of neighbor higher than link delay, the priority queue with current computed distance to the controller is updated and n is set as the predecessor of the neighbor node. After setting the graph and port, the controller calculates the loop, if neighbor nodes have no loop then the Path (P) is selected and sends the data using the selected path.

A picture containing graphical user interface

Description automatically generated

Where the PN denotes predecessor nodes and n denotes the number of nodes. The complexity improves with priority queue and denotes with 0 (L+ n. logN), the L is a link between switches. When PN reaches its maximum value, the overall asymptotic complexity of the sent message considers 0 (N\_2). The n directly connected to controller where EN is empty node and log N basically means time goes up linearly while the n goes up exponentially. The controller load monitoring, the number of DM P-O and P-I messages need to send or receive. The MAC address in destination MD message is one of the multicast addresses.

A picture containing schematic

Description automatically generated

Where the switches(n) denote the set of switches connected with a link and the total P-O messages send by the controller to discover the network and link between switches with active port P for switch i as Pi. The proposed method is more efficient due to the higher number of ports for the network.

**Diagram

Description automatically generated**

**A picture containing text, person, screenshot, document

Description automatically generated**

### 1.2 Flow Information Detector (FID)

This module is used to collect the flow information and report the switch which has packet counter related information, all connection entries and reporting or predecessor switch entries. This information is store in the repository. At the same time, the controller receives all information on switch related to processes and any unusual packets behavior. The malicious switch also changes the path and drop the packet. Whenever the data packets initiate the wrong path, the predecessor switch matches the entries. If there is not any entry exist, then switch reports to the anomaly detector system controller.

## 2. Detection

After topology discovery and active and backup paths are computed, the next phase is malicious node detection. To check the malicious node, this module uses packet flow, topology, switches and link failure information of the entire network. This phase processes discussion in the next sub-sections.

### 2.1 Malicious Traffic Detector (MTD)

The controller acts as an IDS and calculates the active and backup paths and process the rules and initiate for all the switches. The links between switches are the main points for attacks where an attacker alters the rules by using eavesdropping on the data plane and controller to switch. This alteration leads to fraudulent insertion for data dropping or wrong data forwarding. When packets received on switches, it placed in flow buffer and apply the rules matching entries against the flow table. If the rule is matched, then the packet is removed from the buffer and then the switch acts to out the packet from the port with priority and time out value. The packet may take an alternate path to reach the correct destination whenever a link fails. The proposed system detects those packets which are changing the path by link failure or attack. Whenever a host sends a request for the forwarding packet flow, the controller initiates the paths and install connection entries along with alternate paths.

When switch S-1 receives the packet flow, then it placed in flow buffer and match with flow entry and initiates action for the port and place timeout values. On the other hand, if the S-1 receives packet Pi while time is not out, then it checks the condition that packet Pi is matched with flow entry or not. If entries match, then forward the packet to the next switch and add increment. If entry is not matched, then report to S-1 and ask SDN-ADS, where to send the packet.

This module collects all the matches about reported switch including packet counter-information, connection entries, and entries of predecessor switch. If all information is not matched, then S-1 declares as a malicious switch and excluded for further traffic forwarding. In case of a link failure, packet flow takes an alternate path and check switch count variable values and update at each switch. When the packet changes the path, the count variable sends to SDN-ADS and sends hello message to S-1 and update from where packet change the path and next S=1+1 will be computed switch. The switch responds the hello message to the controller.

**Diagram

Description automatically generated**

**Text

Description automatically generated**

### 2.2 Isolation Module

This module isolates the malicious detected switch by matching it with the repository in the next network topology detector process. Whenever any packet initiates the wrong path then the IDS controller calculates the switch count and declares the malicious behavior. If any link fails, then the packets initiate the wrong path but reach to the correct destination and lead to packet delay and sometimes dropped because of timeout.

### 2.3 Storage Module

This is a repository where all the collected and calculated information is store. In case of any error reported from the IDS controller, then it matches with store repository and with installed entries.

# Predicting Network Attack Patterns in SDN using Machine Learning Approach (Saurav Nanda, Faheem Zafari, Casimer DeCusatis, Eric Wedaa and Baijian Yang)

In this paper, we apply four different machine learning algorithms: C4.5, BayesNet (BN), Decision Table (DT), and Naive-Bayes (NB) to predict the potential vulnerable host that might be attacked, using historical network attack data.

* The first paper that leverages Machine Learning approach for defining security rules on the SDN controller
* We compare and evaluate the performance of four widely used ML algorithms. However, our goal is to show the viability of ML approach in SDN security rather than highlighting the four algorithms used.
* We show that even a small probability of attack, obtained through ML approach, has significant effect on the SDN security.

## Machine Learning Algorithms

1. **C4.5 Decision Tree:** C4.5 Decision tree is widely used for inductive inference. In C4.5, the discrete-valued functions are approximated, and decision tree is used for representing the learned function. C4.5 is based on heuristic hill climbing and carries out non-backtracking search throughout all possible decision trees. In C4.5, the data is partitioned into subgroups recursively. C4.5 is robust to highly noisy data and is preferred for learning various disjunctive expressions.

Steps

• An attribute is selected, based on which a logical test is formulated.

• Each test outcome is used as a branch and subset of the training data that satisfies the outcome is moved to the corresponding child node.

• The process is run recursively on all the child nodes.

• A leaf is declared as a node based on specific termination rule.

1. **Bayesian Network:** Bayesian Network or Bayes Net encodes probabilistic relationships among different variables of interest. It consists of a number of variables and set of edges between the variables, resulting in an acyclic graph. Every node in the graph represents the random variable and a directed edge from one variable to another. Every variable in the Bayesian network is independent of the nondescendants. Bayes Net has been used as classifier and if trained properly, can result in highly accurate classifications.
2. **Naive-Bayes:** Naive-Bayes uses Bayesian theory that predicts the type of the unknown samples based on prior probability using the training samples. The Bayesian classification model relies on statistical analysis and Bayesian theory that consists of the Bayesian learning. Bayesian learning uses the prior and posterior probability in combination and uses it to find the posterior probability as per the supplied information and data samples. The Naive-Bayesian algorithm operates by segregating the training set into an attribute vector and a decision variable. The algorithm also assumes that every member of the attribute vector independently acts on the decision variables.
3. **Decision Table:** Decision Table (DT) is used to organize and document logic in a way that assists in easy inspection. DT helps in representing the machine learning output as the input, and involves selecting some of the data attributes. They also assist in evaluating different set of rules for ambiguities and redundancy.

We use machine learning (ML) algorithms to predict potential target host attacks based on the historical network attack data for SDN.

Accuracy = Number of correctly predicted attacks / Total number of attacks ×100

While testing, we also chose a threshold level α percentage as the minimum probability required to consider any host as vulnerable. We altered the values of α to evaluate its effect on the classification accuracy.

### Algorithm 1 Machine Learning based predictor for SDN network attacks

1: procedure ML-BASED ATTACK PREDICTOR

2: Chose the Machine Learning Algorithm

3: Train the ML algorithm using historical data

4: if the trained model predicts an attack on a host by an IP then

5: Update the SDN controller rules to block the IP subnet

6: else Allow the IP to access the resources

Datasets Size

1 278,598 (With Chinese attack data)

2 187,488 (Without Chinese attack data)

3 91,110 (Only Chinese attack data)

Effect of α on average prediction accuracy

α (%) Avg. Prediction Accuracy

0 97.06

1 95.78

5 85.74

10 75.59

The increase in α from 0-10% reduces the prediction accuracy by 21.47%.

With the four different algorithms, and it can be seen that highest average prediction accuracy of 91.68% is attained with BayesNet.

## Effect of the Dataset

The dataset also plays an important role in the prediction accuracy. The higher the variance in data, the higher will be the chances of false prediction. Since dataset 2 did not have the entries from the Chinese attackers, the variation in the dataset was much lesser than dataset 1 and dataset 3. The average prediction accuracy is higher for dataset 2 when compared with dataset 1 and dataset 2.

# Hybrid Deep Learning-based Anomaly Detection Scheme for Suspicious Flow Detection in SDN: A Social Multimedia Perspective (Sahil Garg, Member, IEEE, Kuljeet Kaur, Member, IEEE, Neeraj Kumar, Senior Member, IEEE, and Joel J. P. C. Rodrigues, Senior Member, IEEE)

ANOMALY DETECTION MODULE

This module consists of two phases:

* feature selection
* classification

1. Flow controller requests the FEs (forwarding elements) via OF (OpenFlow) protocol to provide flow statistics.
2. The flow collection module of the controllers collects this information to extract the flow features. Based on this, dimensionality reduction is performed using improved RBM algorithm.
3. The extracted features are passed to the next phase which pre-processes the flow features and performs classification on the network flows with the proposed Gradient based SVM algorithm.
4. The anomaly detection scheme then generates an anomaly report and sends the same to the SDN controller with the help of a secure channel(s).
5. SDN controller updates the flow table in accordance with the report received and configures OF-enabled FEs for further treatment.

### **Algorithm 1** Working methodology of the proposed scheme

1: **procedure** SUSPICIOUS\_FLOW\_DETECTION

2: A user initiates a social multimedia request over Internet

3: Flow controller captures flow statistics

4: Flow features are extracted

5: Dimensionality reduction using improved RBM

6: Classification using Gradient based SVM

7: Anomaly Detection Module generates anomaly report

8: Anomaly report is transmitted to the SDN controller over secure channel

9: for ∀ flows do

10: if flow is anomalous then

11: Discard the associated packet

12: else

13: Apply MoFR to establish an optimal route

14: Controller makes flow table updates

## Dimensionality Reduction: Restricted Boltzmann Machine

RBM is a stochastic approach which learns a probability distribution over the input. It consists of two layers of binary units: one visible, to represent the data, and one hidden, to increase learning capacity. In the proposed approach, RBM is used for dimensionality reduction.

**Algorithm 2** Dimensionality Reduction using RBM

**input:** Training dataset with flow features received from FEs

**output:** Extracted features F = {f1, f2, · · ·, fk}

1: Load training dataset

2: Sample training vector from training dataset

3: Initialize weights W and bias a and b

4: Set m visible units (v)

5: Set n hidden units (h)

6: Compute conditional probability P for all v

7: Compute conditional probability P for all h using dropout

8: Initialize target class c = {c1, c2, · · · , ct}

9: Set training objective as OT = − PT t=1 log P(ct, vt)

10: To deal with the computational problem, compute gradient of log P(ct, vt), i.e., ∂ log P(ct, vt)/∂θ

11: Compute contrastive divergence from gradient to yield features from T

12: Compute Expectation with respect to the data distribution, i.e, E[v] using Gibbs sampling

13: Repeat the procedure G times to obtain a G-steps Gibbs samples

14: Return extracted feature set F

## Classification: Support Vector Machines

The proposed anomaly detection scheme employs SVM for the classification of network traffic flows. It is a supervised learning approach which maximizes the geometric margin between two classes in n-dimensional space.