Department of Electrical and Computer Engineering

Senior Design Project

Software Defined Network (SDN) Intrusion Detection using Machine Learning

Submitted By

Name	ID
Samiha Islam	1931393642
Muhammad Zawad Mahmud	1931401042
Md. Solayman Hossain	1931565042

Submitted To

Dr. Mohammad Monirujjaman Khan

Associate Professor

Department of ECE

North South University

1. Background Study

Using software-based controllers or application programming interfaces (APIs) to communicate with the network's underlying hardware architecture and control traffic is known as software-defined networking (SDN). This architecture is distinct from conventional networks, which manage network traffic using specialized hardware (such as switches and routers). SDN can use software to build and manage virtual networks or manage conventional hardware. SDN decouples the software from the hardware, much like any other virtualized technology. SDN keeps the hardware in charge of the data plane, which actually delivers the traffic, while moving the control plane, which decides where to send it, to software. Through the use of software-defined networking, this enables network managers to program and manage the entire network from a single interface rather than device by device. Now many people are behind trying to get control of SDn systems. It will allow them to connect to all the networks and also access many data. To avoid this, SDN intrusion plays part. If we can predict an attack early before it even take place then we can save many data loses.

With the use of machine learning (ML), which is a form of artificial intelligence (AI), software programs can predict outcomes more accurately without having to be explicitly instructed to do so. In order to forecast new output values, machine learning algorithms use historical data as input.

Machine learning is significant because it aids in the development of new goods and provides businesses with a picture of trends in consumer behavior and operational business patterns. A significant portion of the operations of many of today's top businesses, like Facebook, Google, and Uber, revolve around machine learning. For many businesses, machine learning has emerged as a key competitive differentiation. ML algorithms will help us to predict any attack early and it will allow us to take necessary action.

2. Description of the problem being solved

In the last few years, big companies have been depending more and more on Software defined Networks “SDN” to fulfil their needs for programmable networks. But like other networks, SDN has some security problems. Many technologies are used to solve such problems, and machine learning is considered one of the best. Machine learning has proved its ability to find data patterns when other technologies failed.

This makes it a perfect choice for anomaly-based detection and intrusion detection system "IDS" in general. Here we are proposing a new anomaly-based IDS that benefits from the ability of SDN to provide statistical features about every flow that passes through the network, and passes these features to a voting system that consists of several machine learning algorithms, which gives the system the ability to study the users' behaviour and predict any possible intrusion.

3. Review of existing similar systems

Vigneswaran and Poornachandran [2], introduce an anomaly-based IDS that works in traditional networks and depends on deep neural network model, they use KDDCup99 dataset to train and test the model. The proposed solution gives an accuracy of 93%. They use KDDCup99 dataset, which suffers from imbalance classes and redundant records which affects the reliability of the results. Ajaeiya and Adalian [3], suggest an anomaly-based IDS that works in SDN and only uses the features provided by it. They compare the results of multiple machine learning algorithms. Random Forest algorithm gives the best results, where the true positive rate is 96.3% and the false positive rate is 0.009. The results show the efficiency of depending on the probabilistic distribution using algorithms like Random Forest. However, in their research, they do not use a standard dataset, which raises some concerns about the validity of their results. Abubakar and Pranggono [4], propose an IDS that works in SDN and consists of a signaturebased IDS, and an anomaly-based IDS that depends on deep neural network model and uses NSL-KDD dataset for training and testing. The detection accuracy is 97.4%. However, intrusions detected by the signature-based part are not separated from intrusions detected by the anomaly-based part.

So, the accuracy of the anomaly-based part cannot be isolated.

4. Objective

The objective of this system is to predict attacks on software defined networks (SDN) early. It will help to prevent this attack. SDN is used by many government sectors of a country and those got many confidential data. As this system will help to predict attack early, so it can prevent many confidential data lose. The accuracy rate is the key goal while using the dataset. As mentioned later in this work, various strategies have been tried to improve accuracy.

5. Feasibility study indicating at least two possible solutions

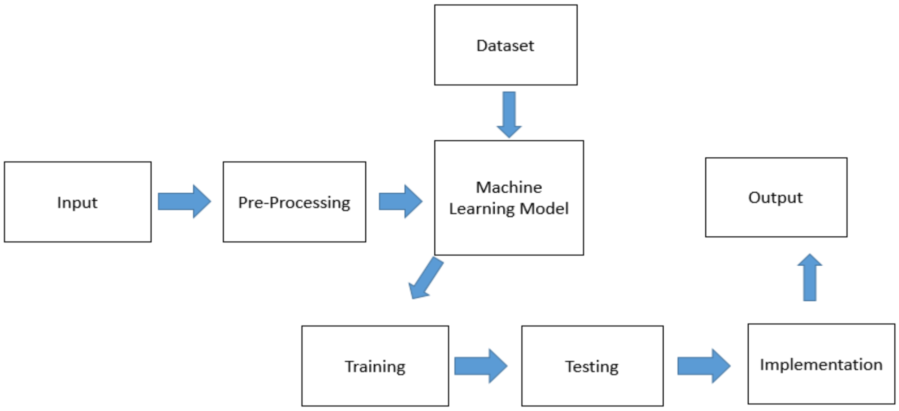
Machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. It explores the construction and study of algorithms that can learn from, and make predictions on data. Machine learning is classified as supervised learning when training data is labeled, unsupervised learning when training data is unlabeled and semi-supervised when training data is a mixture between labeled and unlabeled data. Many machine learning algorithms have been developed and improved in the last two decades, from these algorithms we will use the following:

- Decision Tree (DT): The algorithm chooses the feature with the highest information gain to be the root node, then the 'gini index' is calculated to find the best partition, then the process is repeated till reaching the specified maximum depth.
- Random Forest (RF): A number of decision trees are built depending on a different subset of the dataset for each of them, then the performance of all the trees is averaged to get the final result of the algorithm.

6. Output

Two machine learning algorithms will be used in this system. The advantage of using these two models is that they provide comparative analysis. These comparisons enable us to determine which model provides the best level of accuracy. Finally, we can determine which system performs better at detecting intrusions.

7. Detailed diagrams for the complete system and all subsystems



8. Explanation of the functioning of the complete system, and all subsystems

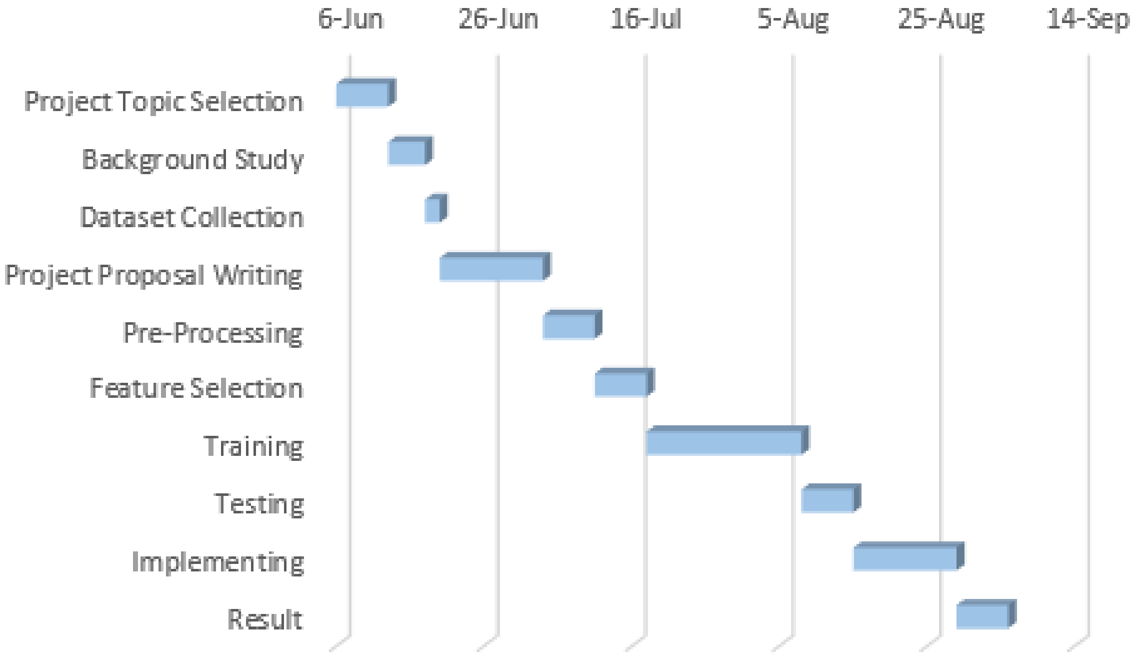
Pre-Processing: Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

Training: Training data (or a training dataset) is the initial data used to train machine learning models. Training datasets are fed to machine learning algorithms to teach them how to make predictions or perform a desired task.

Testing: The testing is set of observations used to evaluate the performance of the model using some performance metric.

Implementation: Implementing machine learning algorithm in untrained data.

9. MS Project charts including Gantt Charts showing the expected timeline of progress



10. Working Steps

- Week-1: Project Topic Selection
- Week-2: Reading the existing papers based on the related topic.
- Week-3: Dataset Collection (https://www.kaggle.com/datasets/subhajournal/sdn-intrusion-detection)
- Week-4: Project Proposal Writing
- Week-5: Pre-Processing
- Week-6: Feature Selection
- Week-7: Training.
- Week-8: Testing
- Week-9: Implementation
- Week-10: Result

11. Major Milestones

- Analyzing the existing system.
- Dataset connection
- Pre-processing the data
- Feature selection
- Training
- Testing

12. Research Methodology

For our prediction part, we will be using 3 popular machine learning algorithms. We are hoping to get the best result out from these. These models are:

1. Support Vector Machine (SVM): The Support Vector Machine (SVM) is a classification and regression Supervised Machine Learning Algorithm. It is most commonly used for classification, but it can also help with regression. SVM basically finds a hyperplane that divides the various types of data. This hyper-plane is nothing more than a two-dimensional line. SVM plots each data item in the dataset in an N-dimensional space, where N is the number of features/attributes in the data. Next, choose the best hyperplane for data separation. SVM performs quite well without any adjustments for linearly separable data. Linearly separable data is any data that can be shown in a graph and divided into groups using a straight line. Kernelized SVM is used for non-linearly separable data. Assume we have some non-linearly separable data in one dimension. This data can be transformed into two dimensions and linearly separated in two dimensions. In this manner, each 1-D data point is mapped to a corresponding 2-D ordered pair. So, in any dimension, we can simply move any non-linearly separable data to a higher dimension and then make it linearly separable. This is a massive and all-encompassing transformation.

A kernel is nothing more than a data point comparison. The kernel function in a kernelized SVM tells you how similar two data points from the original feature space are to points from the newly converted feature space.

There are numerous kernel functions available, but two of the most common are:

Radial Basis Function (RBF): As illustrated below, the distance between the vectors and the original input space is an exponentially decreasing function of the similarity between two points in the modified feature space. RBF is the default kernel in SVM.

Polynomial Kernel: The degree parameter of the Polynomial kernel influences the model's complexity and transformation computing cost.

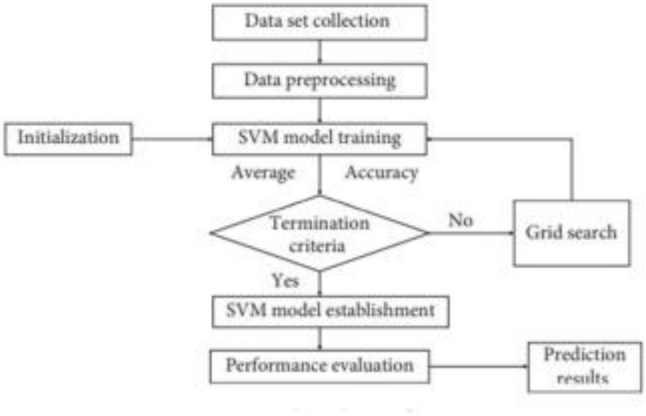


Figure: Support Vector Machine Classifier block diagram

2. Random Forest: Random forest is a machine learning classifier which consist of a collection of tree structured classifiers

$\{h (x, \Theta k), k=1\ldots\}$ - - - - - - - - - - - - (2)

From (2) {Θk} represents random vectors distributed independently identical and each tree has a vote for the most famous class at input x. The nature and dimensionality of Θ depends on its use in tree construction (Breiman, 2001). A number of decision trees are built depending on a different subset of the dataset for each of them, then the performance of all the trees is averaged to get the final result of the algorithm.

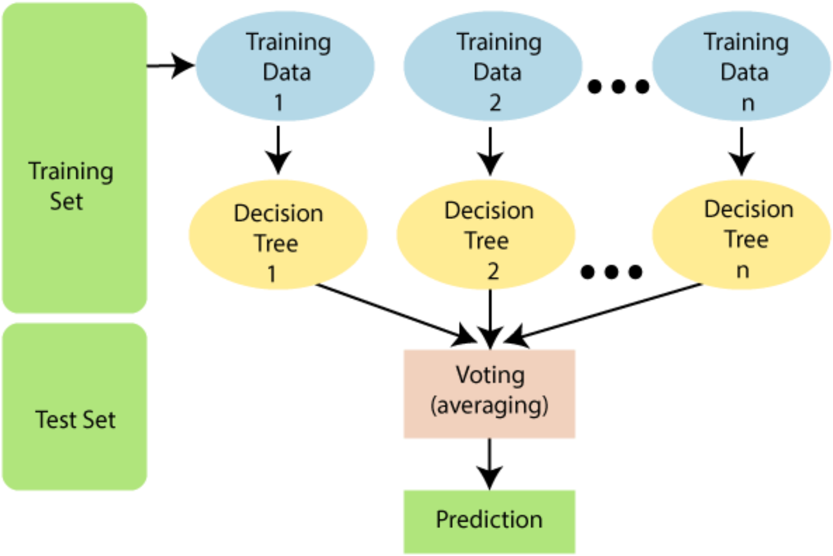


Figure: Random Forest Block Diagram

3. Decision Tree (DT): The algorithm chooses the feature with the highest information gain to be the root node, then the 'gini index' is calculated to find the best partition, then the process is repeated till reaching the specified maximum depth.

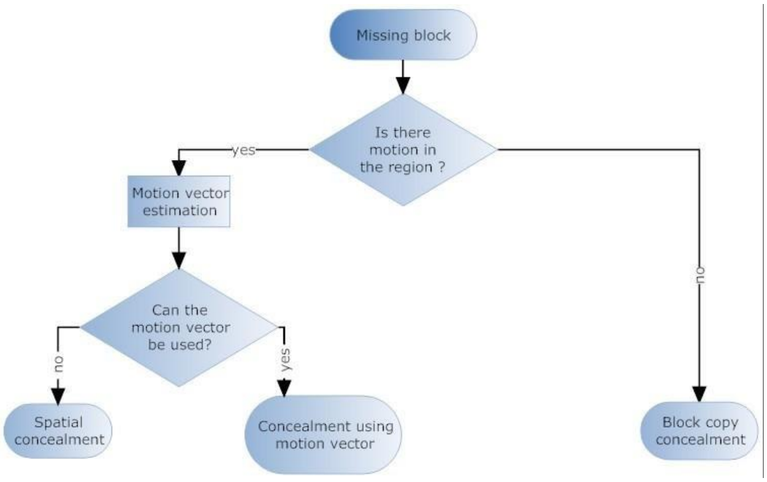

Missing block
Is there motion in the region ?
yes
no
Motion vector estimation
Can the motion vector be used?
no
yes
Spatial concealment
Concealment using motion vector
Block copy concealment

Figure: Decision Tree Block Diagram

13. Required software tools

The model training and validation will be done using Anaconda Navigator and Jupyter Notebook. And Python will be the programming language we use.
A Python programming language distribution called Anaconda aims to make package deployment and administration in scientific computing simpler (data science, machine learning applications, large-scale data processing, predictive analytics, and so on). The version includes data-science applications for Windows, Linux, and macOS.

Using the web application Jupyter Notebook, you can create and distribute documents that include text, images, and code. It is helpful for a wide range of tasks, including data science, statistical modeling, machine learning, and more. Because Jupyter Notebook can connect to numerous kernels, we wish to use this. A program that reacts to various requests (such as code execution, code completions, and inspection) is a Jupyter kernel.

Python is a high-level, dynamically semantic, interpreted programming language. Due to its high-level built-in data structures, dynamic typing, and dynamic binding, it is ideal for rapid application development as well as for use as a glue language or scripting language to connect pre-existing components. Python's clear, simple syntax prioritizes readability, which reduces the cost of software maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

14. Target Population

The people of Bangladesh are our target audience. People in Bangladesh are largely unaware of the attacks which can be made in a software defined networks. As a result the government can lose a lot of confidential data. Through our efforts, we will be able to assist the citizens of our nation in comprehending what it is and how it affects them. If we can predict the attack before to that or in a short period of time, many data can be saved.
As a result of our research, we anticipate being able to identify the attacks on SDN before it even took place or at an early stage.

15. What makes the solution an ‘innovation?’

All the related papers shows accuracy while using Bayes Net, Random Tree and couple of such algorithm is below or close to 90%. Our dataset is an efficient dataset, so we will try to increase the accuracy.

Our key goal was to not only correctly diagnose depressed users, but also shorten the time it took to predict their state.

16. Sustainability of the project

Yes, this is a long-term project because it is a software project written in Python that is easy to understand and maintain, unlike hardware projects

17. Project Scalability

The project can be expanded to handle a bigger number of datasets. This will enhance the accuracy rate even further. It can also be taught to recognize various types of attacks on SDN. If scaled up and other types of attack predictions are added, this project might become the one-stop shop for all potential SDN attacks.

18. Income Generation

The model can be utilized by big software companies to improve their prediction of attacks of SDN using the techniques used, such as trained models and algorithms. Human errors are less likely when the model does all of the forecasting. This might be an intriguing concept that encourages users to employ prediction algorithms in this way. It can also be used by government or offered to the general public, if the market is ready to accept it.

19. Funding

Currently, no funding is required to complete the analysis for this study.

20. Benefit

This system will help many people who are using software defined network but not aware of what problems can be caused by this. There are high possibilities of data lose as anyone can take full control by only taking the controller under control. Our system will provide security by predicting the attack before or at earliest stage and prevent data lose.

21. Risk Factor

We have no risks except that after implementing the algorithms, we may yield less accuracy.

22. Environmental Impact

There is not much environmental impact as it is a software based system.

23. Existing Research Publications

Paper Name	Dataset Name & Link	Algorithm	Accuracy (%)
An Intrusion Detection System Using Machine Learning Algorithm	The KDD Cup 1999 data set	Bayes Net J48 Random Forest Random Tree	86.1 96.2 97.7 97.4
Evaluation of Machine Learning Algorithms for Intrusion Detection System	KDD intrusion dataset	MLP Random Tree Random Forest J48 Naive Bayes Bayes Network Decision table	91.9 90.57 93.77 93.1 91.23 90.73 92.44
Enhancing Network Intrusion Detection Model Using Machine Learning Algorithms	KDD99 dataset	Naïve Bayes Decision Trees K-Nearest Neighbor Decision Table	90 99 94.6 98.5
Machine Learning based Intrusion Detection System for Software Defined Networks	KDD99 dataset	Decision Tree Random Forest XGBoost Support Vector Machine	99.4 98.54 99.05 89.68
Machine Learning based Intrusion Detection System for Software Defined Networks	NSL-KDD	Decision Tree Random Forest XGBoost Support Vector Machine	82.72 77.40 79.61 73.22

24. Conclusion

SDN which is the short form of software defined network is one of the most populist model. As it is becoming popular, unethical people are giving their eye on it. This model has a controller which controls the entire network. Hackers plan to take control of the controller and access many information. Our system will take help of various machine learning algorithm to predict the attack before it takes place or at earliest stage. As a result many information will be saved.

25. Bibliography

[1] Oqbah Ghassan Abbas, Khaldoun Khorzom and Mohammed Assora, "Machine Learning based Intrusion Detection System for Software Defined Networks"

[2] R. Vigneswaran and P. Poornachandran, "Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security," in 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, 2018.

[3] G. Ajaeiya and N. Adalian, "Flow-Based intrusion detection system for SDN," in 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, 2017.

[4] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," in Seventh International Conference on Emerging Security Technologies (EST), Canterbury, 2017.

[5] Machine Learning based Intrusion Detection System for Software Defined Networks – IJERT

[6] https://www.techopedia.com/definition/14650/data-preprocessing

[7] https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/

[8] https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html

[9] https://www.kaggle.com/datasets/subhajournal/sdn-intrusion-detection

[10] https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/