



Intrusion detection in secure shell using Multilayer Perception

Er. Amit Chapagain*

Department of Computer Science, Academia International College

Kathmandu, Nepal

aamitchapagain@gmail.com

Original Article

Correspondence Author*

Received: Jul 07, 2026

Revised & Accepted: Aug 19, 2026

Copyright: Author (2026)



This work is licensed under a Creative Commons Attribution-Non-Commercial 4.0 International License

ABSTRACT

As the use of Internet users increases, their inter-connectivity enables malicious users to exploit resources and surge Internet attacks. The increasing Internet attacks pose various difficult challenges to develop flexible, adaptive, reliable security focused approaches. Intrusion Detection System (IDS) is the most essential components being used to detect Internet attacks. Intrusion Detection System is a system which monitors, analyzes and detects the events that are considered as violation to the security policies of a networked environment. Even Encrypted protocols are violated, Secure Shell server can be an appropriate repository to launch Brute force attacks, distribute spam messages, and assess new malware. The traffic of encrypted protocols like SSH makes packet payload examination challenging and slow. In flow monitoring techniques, flows in aggregated network data are observed. Data flow is a set of packets that passes in a certain time interval and that has a similar set of attributes. CICIDS2017 is used in this research of Intrusion Detection techniques. Analysis of data set with respect to four classes which are BENIGN and SSH-Patator in which all data attributes can be categorized. A predictive model which uses an idea of



detecting intrusion in network is capable of recognizing intrusions or attacks as “1”, and normal connections as “0” using Multilayer Perceptron (MLP) classification.

Keywords: Intrusion Detection System, Machine Learning, Secure Shell, Multilayer Perceptron

1. INTRODUCTION

The rapid increase in individuals utilizing and depending on the Internet makes breaching into systems, further beneficial interest for Intruders. An intrusion detection system (IDS) is like a watchdog, that monitors traffic of network and issues alarms when suspect activity is detected. The example of host-based intrusion detection system (HIDS) observes critical files of operating system, while a system that analyzes incoming network traffic is an example of a network intrusion detection system (NIDS). There are basically two approaches of detection in IDS by detection; In signature-based detection traffic with irregular patterns are recognized as malware whereas anomaly-based detection relies on machine learning algorithm for detecting changes from a model of "good" traffic. Reputation-based detection, i.e., recognizing potential threats based on reputation scores, is another approach. Intrusion detection systems can utilize to understand malicious traffic by utilizing some special tools like honeypots that catch both normal and malicious traffic. (Mohammed & Rehman, 2015).

Most current IDSs take time for detection, which reduces their efficiency. An efficient and trustworthy algorithm should pick the features by making the process to detect bad traffic, and this is an important part of how intrusion detection systems work (Sadek et al., 2013).

Secure Shell (SSH) daemons are a highly targeted class of services, as exploiting them enables an intruder to gain remote host access or full control. Therefore, intrusion detection in SSH is very important for stopping the network and its nodes.

IDSs are mostly based on inspection of packets. In fast speed networks where the throughput is more than 1 Gbit/s, packet payload is not viable. In encrypted protocols like SSH, it is more challenging for inspection of packets (Koch & Rodosek, 2010). In Flow monitoring systems, for example Cisco's NetFlow (Claise, 2004) and the modern IPFIX (Sadasivan et al., 2009) protocol, provide aggregated network data by means of flows.

Flow data can be used in various application areas, including network protection though it basically provides combined view of activities in network (Sperotto et al., 2010). Consequently, flow-based technologies are indispensable for creating NIDS for networks with high speeds. In this context, Sperotto et al. have studied brute-force attacks in SSH and discovered ssh characteristics in level of flow (Sperotto et al., 2009). The study has highlighted three phases of brute force attacks in SSH, which are discovered by their characteristics of data flow.



For this research, I have used Supervised machine learning, which builds a mathematical model, a Multilayer Perceptron (MLP), from data containing both inputs and anticipated outputs. The data used to train the model is called training data. Objective functions are used to predict outputs for fresh inputs through iterative optimization by Supervised learning algorithms. An ideal function enables the procedure to accurately select the output for test data which is excluded in the training data. A model is considered "learned" when it gradually advances the accuracy of its predictions or outcomes for a given job over period.

2. LITERATURE REVIEW

2.1 Related works

For finding out the characteristics of packets generated by various application protocols, the flow based, and packet-based data are sufficient. To distinguish encrypted traffic various research has focused on approach known as data mining.

Bacquet et al. employed a Multi-Objective Genetic Algorithm to discover encrypted network traffic by choosing best flow feature subspaces and clustering parameters (Bacquet et al., 2011). Their methodology is calculated using a private dataset collected from a university campus. SSH is used as the benchmark for encrypted traffic, and packet payloads serve as the ground truth. Building on earlier research display that overall accuracy is meaningfully affected by feature selection and the number of clusters. Authors design the genetic algorithm with four main objectives: maximizing cluster cohesiveness, enhancing cluster separation, minimizing the number of clusters, and reducing the total number of flow features used.

Arndt and Zincir-Heywood investigated the categorization of encrypted traffic by comparing three approaches: C4.5, k-means clustering, and a Multi-Objective Genetic Algorithm (MOGA) (Arndt & Zincir-Heywood, 2011). They used the SSH protocol classification as a case study. The authors highlighted the accuracy and robustness of the algorithms. To evaluate their methods, they tested stability using three public and private datasets. Furthermore, they explored various flow export settings. In total, 46 flow features were taken for evaluation, although each algorithm utilized a various subset of these features.

Alshammari and Zincir-Heywood have published various papers (Alshammari & Nur Zincir-Heywood, 2007) (Alshammari & Zincir-Heywood, 2011) focused on classification of traffic based on various supervised machine learning models. The authors aimed to identify SSH, Skype, and Gtalk traffic using flow features without relying on port numbers, IP addresses, or payload data. Basically, they used 22 flow features, with some studies employing genetic programming for feature collection. Their research involved both public datasets, where ground truth was based on port numbers, and private datasets generated on a test-bed network, labeled with a commercial packet classification tool. They evaluated



several classification algorithms, including AdaBoost, RIPPER, Support Vector Machine, Naive Bayes, Genetic Programming, and C4.5.

R. Sadre and A. Pras (Hellemons et al., 2012) suggest SSHCure, an intrusion detection tool precisely designed to find out SSH intrusion based on flow. It employs effective algorithms for real-time classification of occurring attacks and recognition of intruded targets. A sample model of this algorithm has been developed, along with a GUI, and designed as a plugin for the standard NfSen supervising tool. Eventually, the detection system's execution has been confirmed using empirical traffic data.

Kumano et al. examined encoded traffic that are used by various real time apps (Kumano et al., 2014). Web portals and bulk transfer flows are encrypted utilizing IPsec and PPTP to generate an evaluation dataset. The authors used the C4.5 and SVM algorithms to categorize applications using these datasets. They then tested classification accuracy with differing numbers of payloads from beginning of flows. Additionally, they assessed the impact of using fewer packets on flow features and suggested early application classification using features that show the minimum change.

Barati et al. investigated Secure Shell (SSH) for brute force attack, and suggest a better approach for detection of intrusion (Barati et al., 2014).

Najafabadi et al. established an aggregation of Netflows to extract the appropriate features for developing machine learning models (Najafabadi et al., 2015). The authors' approach to defining discriminative features by considering domain information and analyzing representative traffic holding normal data like the attack data can be used to build machine learning-based models for detecting other types of intrusion as well.

Sadek et al. emphasize how attribute selection for dimensionality reduction influences the development of an effective intrusion detection system that gets high detection accuracy and reduces false alarm rates. (Sadek et al., 2013).

Vinayakumar et al. evaluated the efficiency of various shallow and deep learning networks for describing and classing encrypted tunnels, with a particular focus on SSH tunnels (Vinayakumar et al., 2017). The findings demonstrate that both ML and deep learning techniques perform well across the dataset, which originates from entirely distinct networks.

2.2 Related Theory

Machine learning (ML) is a branch of artificial intelligence. It implies a precise study of procedures and statistical methods that enable computer systems to execute specific tasks efficiently. The ML model is based on patterns and interpretations as an alternative of explicit instructions. ML algorithms create mathematical models using experimental data, called "training data," to make estimates or decisions,

instead of directly instructed to perform the task. Various ML algorithms are classified based on their approaches, types of data they handle, and the specific difficulties they aim to solve.

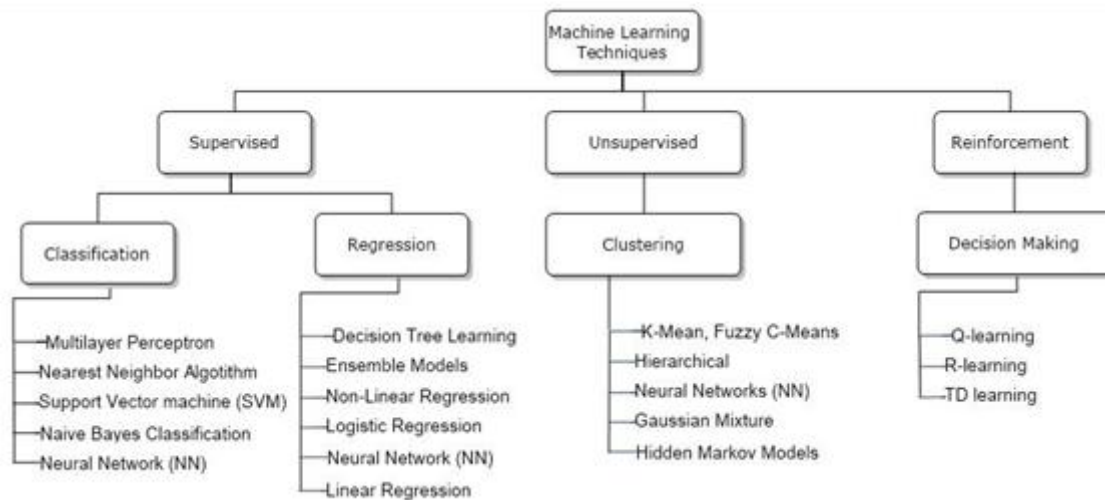


Figure 1: Types of Machine learning algorithms

2.3 Artificial Neural Network

Artificial neural networks (ANN) are motivated by the organic nervous systems and astrocytes composed of brains like animals. The neural network itself is a framework for various ML algorithms to perform tasks collectively and operate on various complicated data inputs. Models "learn" to work by evaluating patterns, commonly by task-specific algorithms instead of hard-coded programs.

2.3.1 Multilayer Perceptron

As a feedforward artificial neural network, the Multilayer Perceptron (MLP) shows the power of advancement that can be achieved using three distinct layers: input, hidden, and output. Individual node, acting as a neuron with a nonlinear activation function, symbolizes brilliance of complexity. MLP apply advanced approach of backpropagation for training. It excels the limitations of a linear perceptron, and unlock the potential to distinguish data by separation.

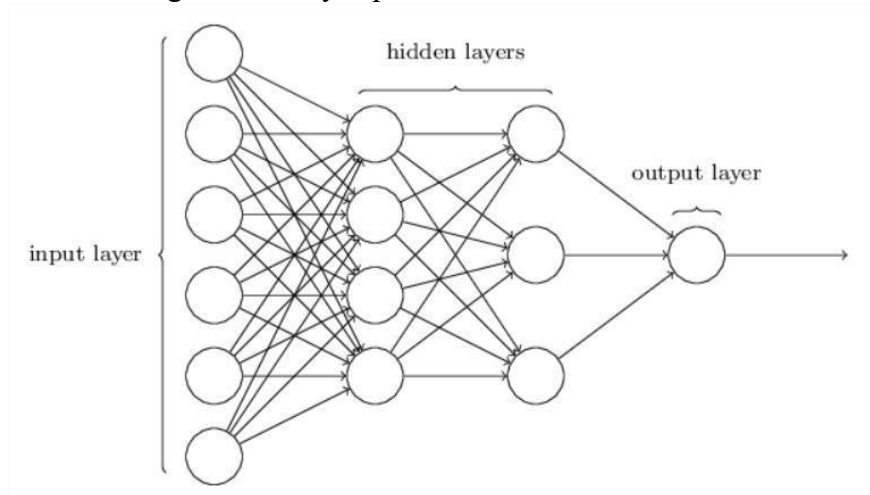




Figure 2: A hypothetical example of Multilayer Perceptron Network

A perceptron is a crucial building block of artificial neural networks (ANNs), dynamic machine learning models inspired by the complexities of the human brain. By mathematically representing a neuron—the brain's fundamental computational unit—the perceptron summarizes the essence of how our nervous system operates. Neurons receive input signals through their dendrites and transmit them along their axons. These axons branch out and create connections with the dendrites of other neurons via synapses, forming a complex network of communication. ANNs mimic this biological structure, enabling them to process information and model intricate associations of inputs and outputs effectively. A single perceptron model acts as a linear classifier. It takes an input vector $x = (x_1, \dots, x_n)$ and produces an output value $f(x)$ based on the following formula:

$$\begin{cases} 1, & \text{if } \sum_{i=1}^m \omega_i x_i + b > 0 \\ 0, & \text{otherwise} \end{cases}$$

Here, “ $w = (\omega_1, \dots, \omega_n)$ ” is weights that have actual value and “ b ” is bias. By incorporating nonlinear transformations ϕ_i into the input vector x , we can enhance the perceptron’s capabilities, leading to more accurate and resilient output calculations for y as:

$$y(x, w) = g\left(\sum_{i=1}^m \omega_i \phi_i(x)\right)$$

In this context, m denotes the number of variables or features used in the model, while g denotes a nonlinear activation function, such as the sigmoid or Rectified Linear Unit (ReLU), which establishes nonlinearity into the model. The perceptron is an essential and straightforward model that can effectively approximate a wide range of continuous functions encountered in real-world data. To improve the model's performance and complexity, we can construct a Multilayer Perceptron (MLP) by incorporating at least one hidden layer beyond the basic perceptron structure. The learning process for this model generally uses the backpropagation algorithm, an advanced version of the least mean squares algorithm originally used to train linear perceptions, enabling it to adjust weights and minimize errors more efficiently.

3. RESEARCH METHODOLOGY

3.1 Research Workflows

For the research of Intrusion detection in SSH, supervised learning technique MLP is used. CICIDS2017 Dataset Collected from Canadian Institute for Cybersecurity is used for detecting intrusion in SSH.

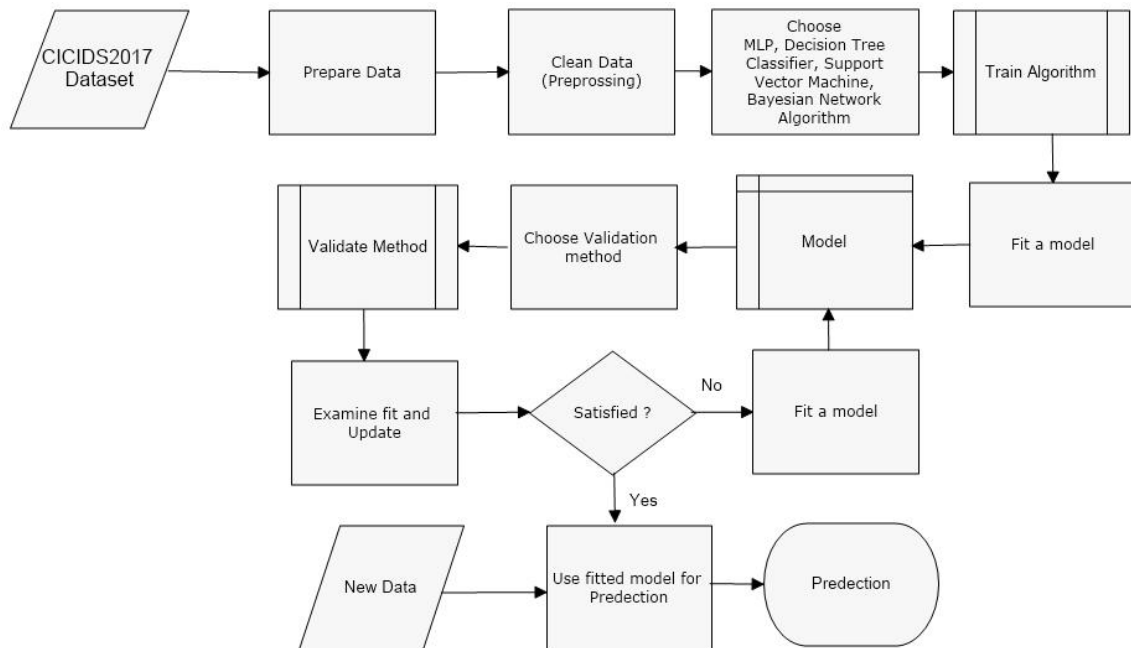


Figure 3: Basic Machine Learning Workflow

3.2 System Model

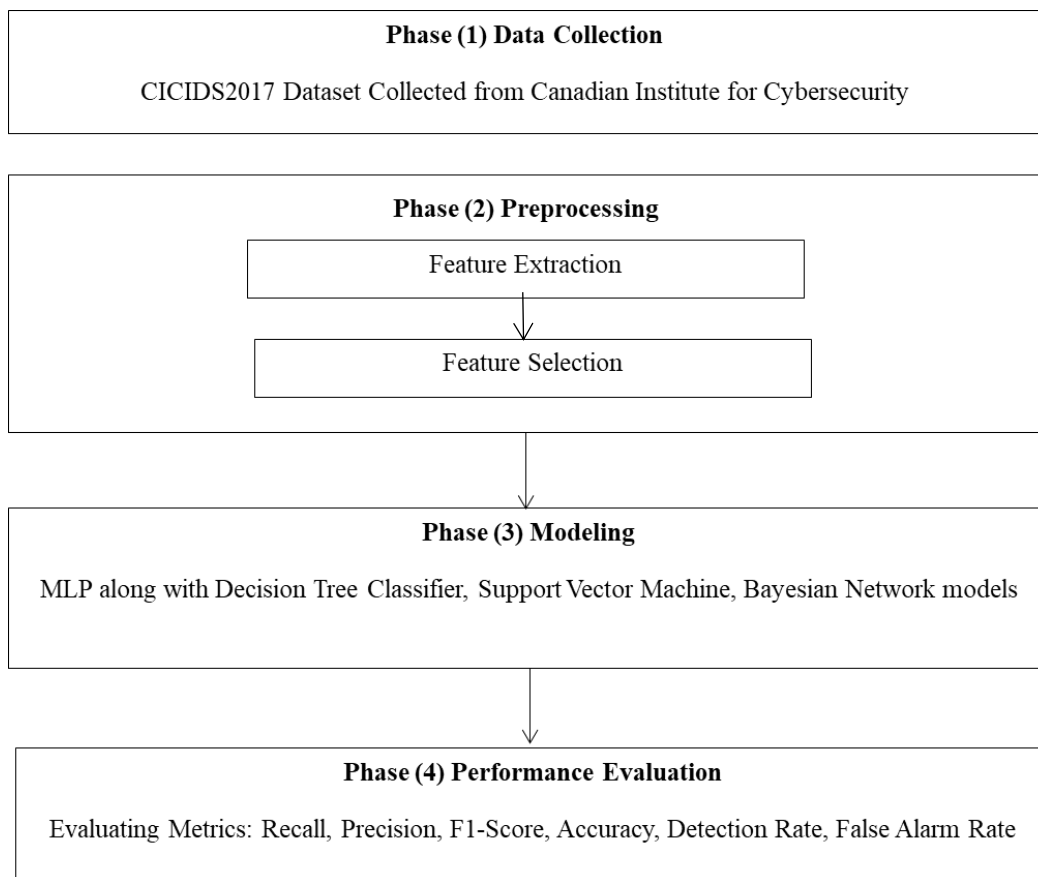


Figure 5: System Model

3.2.1 Phase 1: Data Collection



CICIDS2017 Dataset Collected from Canadian Institute for Cybersecurity. It contains benign and the most common attacks. Used Tuesday, July 4, 2017, dataset which is of Brute Force having FTP-Patator and SSH-Patator and Normal Traffic as Benign for categorial feature with other 78 features.

Percentage of Patator is 1.346% and Percentage of Benign is 98.653%. Added patator data into each divided 5 equal chunks of Benign data now the Patator is of 6.388%.

Table 1: CICID 2017 Dataset Traffic type with description

Traffic Type	Size	Description
Benign	13835	Normal traffic behavior
FTP Patator	7938	The attacker uses FTP Patator in an attempt to perform a brute force attack to guess the FTP login password
SSH Patator	5897	The attacker uses SSH Patator in an attempt to perform a brute force attack to guess the SSH login Password
Total	445909	Total number of Lable including Bening, FTP Patator ans SSH Patator

3.2.2 Phase 2: Data Preprocessing

Feature selection and feature extraction, also called feature construction. Feature extraction is used to extract the features from original dataset that have minimum linear dependencies. Feature selection removes irrelevant features processes for data representation and classification. Variance threshold of 0.1 processed 63 features, then Filter (correlation of 0.85, reduces features to 31) and Wrapper feature selection method (linear regression reduces features to 12) are used, respectively.

Table 2: Final Features selected for the Experiment

1. Fwd Packet Length Min	7. URG Flag Count
2. Fwd Packet Length Max	8. Down/Up Ratio
3. Bwd Packet Length Min	9. act_data_pkt_fwd
4. Min Packet Length	10. Init_Win_bytes_forward
5. PSH Flag Count	11. Total Length of Fwd Packets
6. ACK Flag Count	12. Label

3.2.3 Phase 3: Modeling

Used Supervised Learning Model: Multilayer Perceptron (MPL). Perceptions are biologically motivated, ML model is the main content of ANN. MLP generally contains minimum of three layers nodes, and they are: an input, a hidden and an output layer. Hidden layer node and output layer node is a neuron that uses activation function that are nonlinear. To train the model it applies technique of supervised learning that



is backpropagation which uses ReLU activation function. MLP uses multi-layers nodes and non-linear activation function which is unique from a linear perceptron. Therefore, MLP classify data which is not linearly distinguishable.

3.2.4 Phase 4: Performance Evaluation

Performance evaluations are determined in terms of the detection rate of the intrusion. To evaluate the MLP model, various parameters of confusion matrix are considered. These parameter metrics are commonly used for various purposes of classification models. They are defined as follows:

- TP (True Positive): Total count of actual attack (positive) classified as attack (positive).
- TN (True Negative): Total count of actual normal (negative) classified as normal (negative).
- FP (False Positive): Total count of actual attack (positive) classified as normal (negative).
- FN (False Negative): Total count of actual normal (negative) classified as attack (positive).

Table 3: Confusion Matrix for model evaluation

	Predicted as "0"	Predicted as "1"
Condition as "0"	True Negative(TN)	False Positive(FP)
Condition as "1"	False Negative(FN)	True Positive(TP)

Various metrics can be determined based on confusion matrix parameters.

Value of positive predictive is known as Precision and is given by:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Rate of true positive is known as Recall and is given by:

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

The percentage of detection rate (DR) is given by:

$$\text{DR} = \text{TP} / (\text{TP} + \text{TN}) * 100\%$$

False Alarm Rate (FAR) is given by:

$$\text{FAR} = [\text{FP} / \text{TN}] * 100$$

Accuracy calculates the proportion of the accurately classified traffic samples to the sum of total traffic samples. Accuracy of Intrusion Detection System is very important to ensure the capability of the system to detect intruders.

$$\text{Overall Accuracy} = (\text{TN} + \text{TP}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$



ROC Curve:

A receiver operating characteristic (ROC) curve is a visual presentation which shows how well a binary classification model performs among all classification thresholds. The plot is produced by graphing the true positive rate (TPR) vs the false positive rate (FPR) at different threshold configurations. TPR is also called sensitivity, signifies fraction of true positives among all actual positives. Conversely, the false positive rate is calculated as 1 minus specificity i.e. the true negative rate.

4 RESULTS

4.1 Experimental Environment

The experiment is performed in data science platform of Python called Anaconda, on the system have Processor of Intel i7-7700HQ CPU of 2.80 GHz with 16GB RAM in 64-bit Windows 10 pro OS.

4.2 Tools used

The tools, programming language and software's that will be used in this thesis work are listed below:

- Python Programming Language
- Anaconda: platform where data scientists deploy AI and machine learning projects.
- Various Libraries like Pandas, numpy, scikit-learn, SNS

4.3 Validation Criteria

The validation will be done by

Stratified K-Fold is cross validation technique that is used for Splitting the dataset. It is a modification of K-Fold. It splits given dataset into k folds while maintaining same percentages of each class label in each fold as the whole set. This cross-validation technique makes a fair splits of training and testing groups, as the dataset has uneven number of normal and attack data.

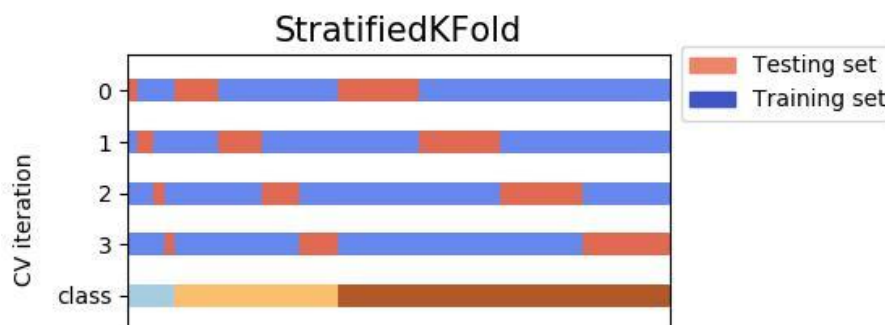


Figure 6: Stratified K-fold Validation

4.4 Heatmap

Heatmap is a graphical model of data that uses a system of color-coding to show the size or amount of values in a data set. It helps in finding patterns and gives a perspective of depth by color values. Generally, Red color indicate high values, while blue color signify low values.

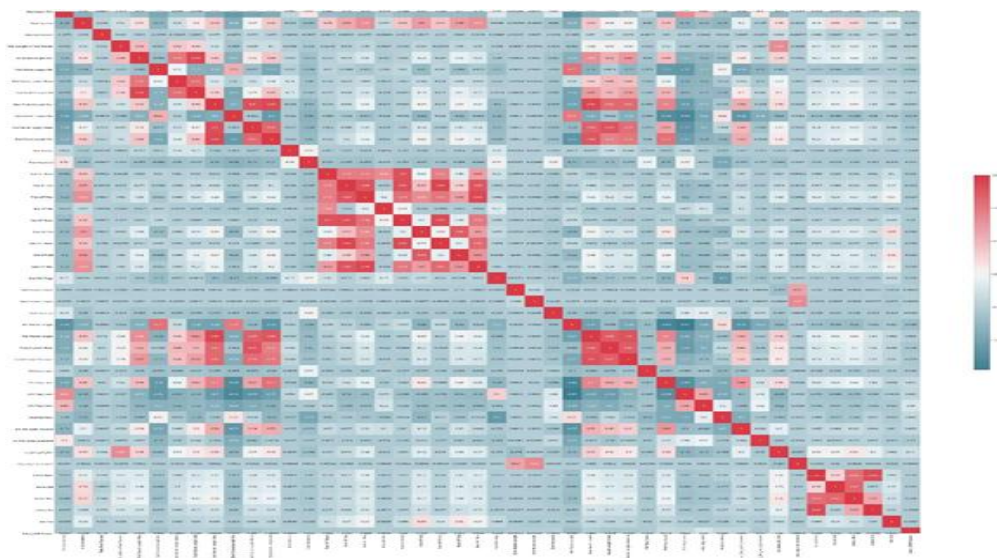


Figure 7: Heatmap



4.5 Experiment using MLP Classifier

In the experiment, MLP classifier with three hidden layers of 10 neurons each is performed with maximum iteration of 1000 and other default parameter is used. The default parameters are

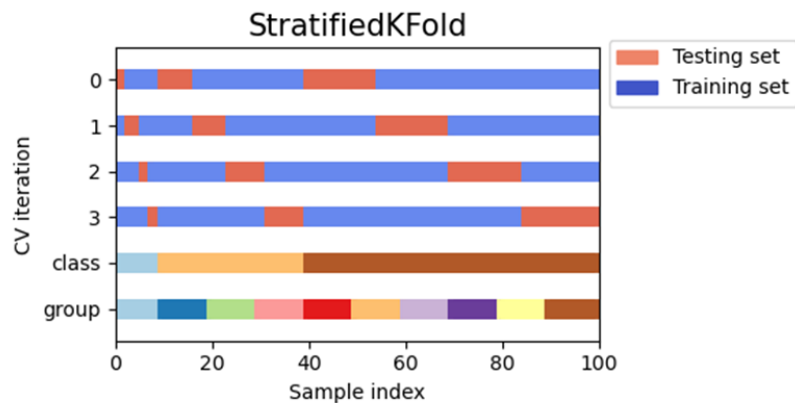
MLPClassifier(hidden_layer_sizes=(50,50), max_iter=200, learning_rate='adaptive')

Table 4: parameters and values used for MLP

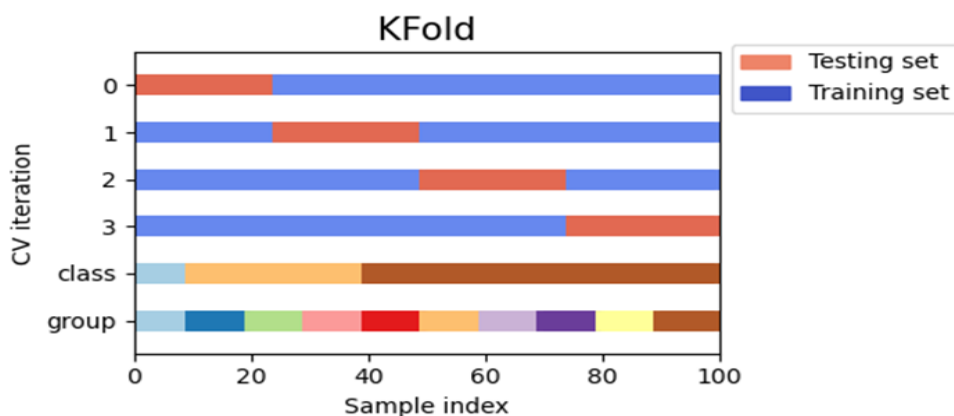
Parameter	Values
hidden_layer_sizes	(50,50)
max_iter	200 (default)
activation	'relu' (default activation)
solver	'adam' (default solver)
alpha	0.0001 (default alpha value)
learning_rate_init	0.001 (default learning_rate_init value)
learning_rate	'adaptive'

Two K-fold Cross Validation (CV) methods are used for each Classifier:

Stratified K-fold CV



Normal K-fold CV



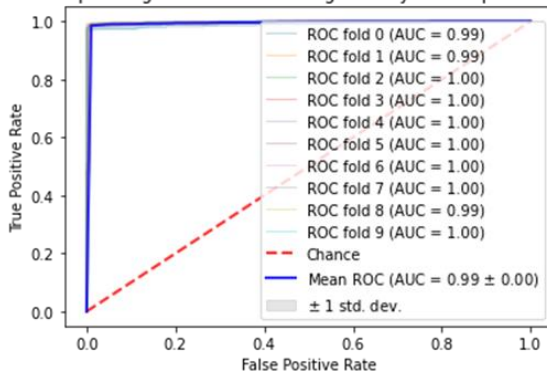


MLP Model Building

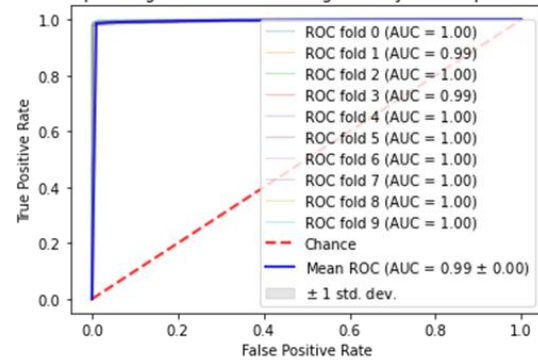
Table 5: Performance Evaluation of MLP using normal K-Fold CV

Performance Evaluation of MLP using Stratified K-Fold CV						
for10 folds	Chunk 1	Chunk 2	Chunk 3	Chunk 4	Chunk 5	Average
Precision	0.9631	0.9668	0.9743	0.9662	0.9707	0.9682
Recall	0.9881	0.9876	0.9654	0.9651	0.9403	0.9693
F1 Score	0.9751	0.9769	0.9688	0.9645	0.9523	0.9675
FAR	0.0057	0.0049	0.0030	0.0040	0.0034	0.0042
DR	0.9820	0.9801	0.9338	0.9343	0.8842	0.9429
Accuracy	0.9935	0.9941	0.9929	0.9919	0.9893	0.9923

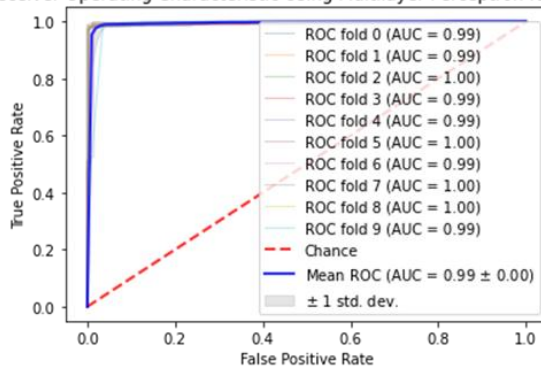
Receiver Operating Characteristic Using Multilayer Perceptron for chunk 1



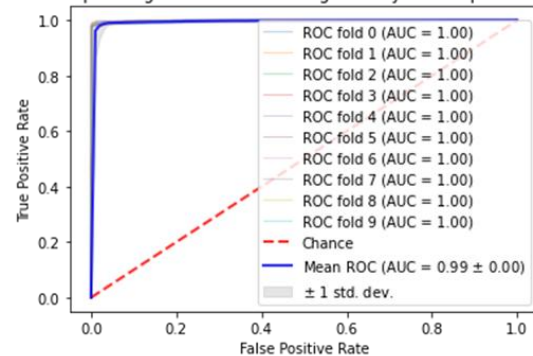
Receiver Operating Characteristic Using Multilayer Perceptron for chunk 2



Receiver Operating Characteristic Using Multilayer Perceptron for chunk 3



Receiver Operating Characteristic Using Multilayer Perceptron for chunk 4



Receiver Operating Characteristic Using Multilayer Perceptron for chunk 5

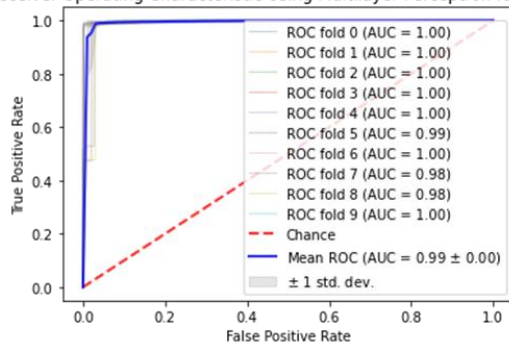


Figure 9: ROC for 5 Chunks of Data for DT using Stratified K-Fold CV



Table 6: Performance Evaluation of MLP using stratified K-Fold CV

Performance Evaluation of MLP using Stratified K-Fold CV						
for10 folds	Chunk 1	Chunk 2	Chunk 3	Chunk 4	Chunk 5	Average
Precision	0.9631	0.9668	0.9743	0.9662	0.9707	0.9682
Recall	0.9881	0.9876	0.9654	0.9651	0.9403	0.9693
F1 Score	0.9751	0.9769	0.9688	0.9645	0.9523	0.9675
FAR	0.0057	0.0049	0.0030	0.0040	0.0034	0.0042
DR	0.9820	0.9801	0.9338	0.9343	0.8842	0.9429
Accuracy	0.9935	0.9941	0.9929	0.9919	0.9893	0.9923

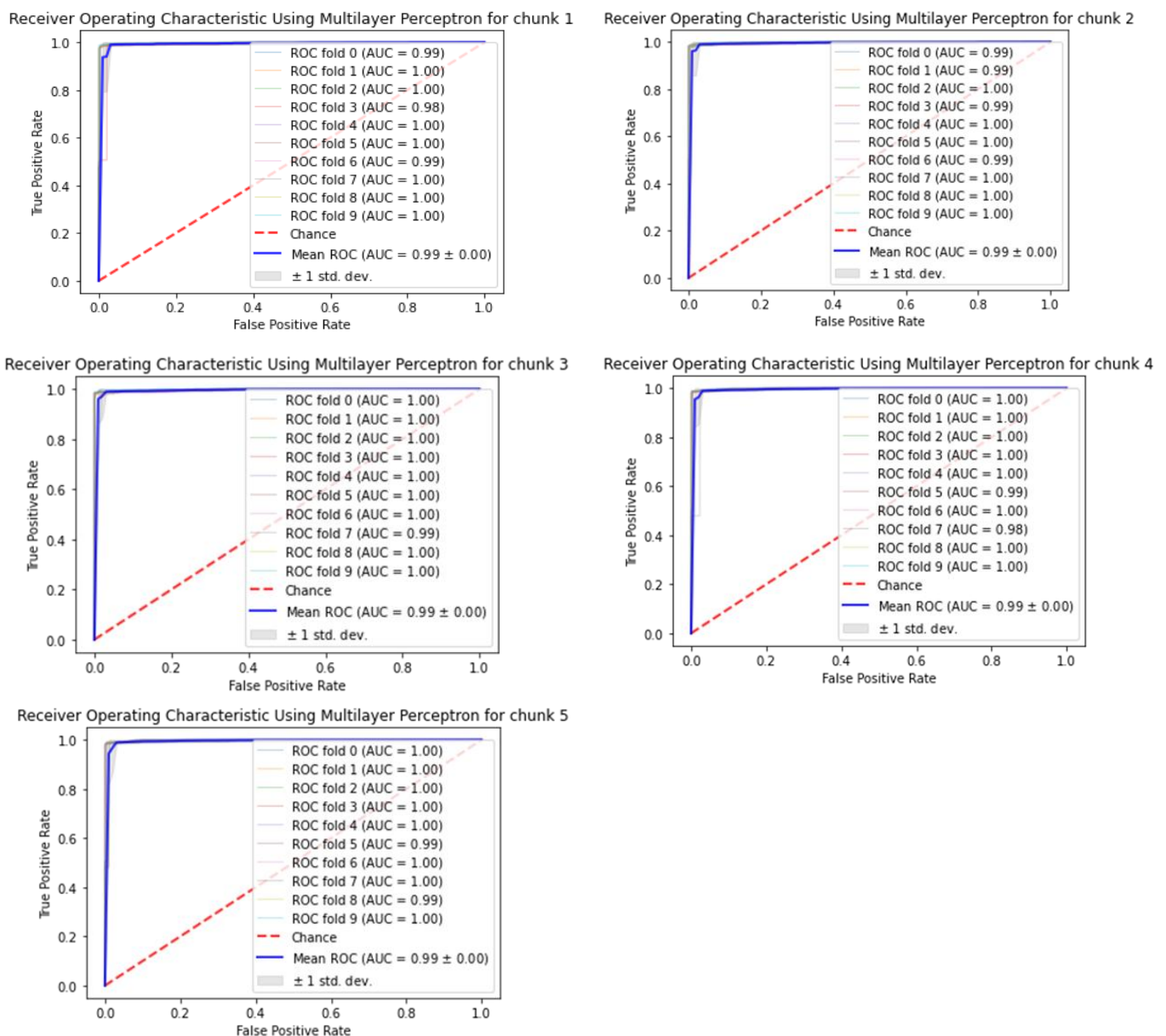


Figure 9: ROC for 5 Chunks of Data for DT using Stratified K-Fold CV



Table 7: Evaluation Metrics Comparison using Stratified and Normal K-Fold CV

	Stratified K-fold CV (%)	Normal K-fold CV (%)
Precision	96.82	95.76
Recall	96.93	96.90
F1 Score	96.75	96.23
FAR	0.42	0.57
DR	94.29	94.38
Accuracy	99.23	99.10

5. CONCLUSION

CICIDS2017 dataset has unbalanced data label, Benign was partitioned into 5 equal partitions and Patator data was added in each partition and formed an equal number of 5 Chunks of Dataset. Each Chunk of data was used in a classification model (MLP) and the average of all 5 Chunks was calculated to generate the best classification model. The feature selection process was performed with Variance Threshold with value 0.01 and is used to select 63 features from 79 total initial features. Filter method and Wrapper method were combinedly used and the results 12 final features.

Hyper-parameters tuning of an MLP Classifier was performed and the best parameters were found. Using Stratified K-Fold CV, even increase the value of Evaluation Metrics for MLP classification model comparatively than that of Normal K-Fold CV.

6. REFERENCES

- Alshammari, R., & Nur Zincir-Heywood, A. (2007). A flow based approach for SSH traffic detection. *2007 IEEE International Conference on Systems, Man and Cybernetics*, 296–301. <https://doi.org/10.1109/ICSMC.2007.4414006>
- Alshammari, R., & Zincir-Heywood, A. N. (2011). Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? *Computer Networks*, 55(6), 1326–1350. <https://doi.org/10.1016/j.comnet.2010.12.002>



- Arndt, D. J., & Zincir-Heywood, A. N. (2011). A Comparison of three machine learning techniques for encrypted network traffic analysis. *2011 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 107–114. <https://doi.org/10.1109/CISDA.2011.5945941>
- Bacquet, C., Zincir-Heywood, A. N., & Heywood, M. I. (2011). Genetic optimization and hierarchical clustering applied to encrypted traffic identification. *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, 194–201. <https://doi.org/10.1109/CICYBS.2011.5949391>
- Barati, M., Abdullah, A., Udzir, N., Behzadi, M., Mahmod, R., & Mustapha, N. (2014). INTRUSION DETECTION SYSTEM IN SECURE SHELL TRAFFIC IN CLOUD ENVIRONMENT. *Journal of Computer Science*, 10(10), 2029–2036. <https://doi.org/10.3844/jcssp.2014.2029.2036>
- Claise, B. (2004). *Cisco Systems NetFlow Services Export Version 9* (Request for Comments RFC 3954). Internet Engineering Task Force. <https://doi.org/10.17487/RFC3954>
- Hellemons, L., Hendriks, L., Hofstede, R., Sperotto, A., Sadre, R., & Pras, A. (2012). SSHCure: A Flow-Based SSH Intrusion Detection System. In R. Sadre, J. Novotný, P. Čeleda, M. Waldburger, & B. Stiller (Eds.), *Dependable Networks and Services* (pp. 86–97). Springer. https://doi.org/10.1007/978-3-642-30633-4_11
- Koch, R., & Rodosek, G. D. (2010). Security System for Encrypted Environments (S2E2). In S. Jha, R. Sommer, & C. Kreibich (Eds.), *Recent Advances in Intrusion Detection* (Vol. 6307, pp. 505–507). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-15512-3_35
- Kumano, Y., Ata, S., Nakamura, N., Nakahira, Y., & Oka, I. (2014). Towards real-time processing for application identification of encrypted traffic. *2014 International Conference on Computing, Networking and Communications (ICNC)*, 136–140. <https://doi.org/10.1109/ICCNC.2014.6785319>
- Mohammed, M., & Rehman, H. (2015). *Honeypots and Routers: Collecting Internet Attacks* (0 ed.). Auerbach Publications. <https://doi.org/10.1201/b19660>
- Najafabadi, M. M., Khoshgoftaar, T. M., Calvert, C., & Kemp, C. (2015). Detection of SSH Brute Force Attacks Using Aggregated Netflow Data. *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 283–288. <https://doi.org/10.1109/ICMLA.2015.20>
- Sadasivan, G., Brownlee, N., Claise, B., & Quittek, J. (2009). *Architecture for IP Flow Information Export* (Request for Comments RFC 5470). Internet Engineering Task Force. <https://doi.org/10.17487/RFC5470>
- Sadek, R. A., Soliman, M. S., & Elsayed, H. S. (2013). *Effective Anomaly Intrusion Detection System based on Neural Network with Indicator Variable and Rough set Reduction*. 10(6).



- Sperotto, A., Sadre, R., de Boer, P.-T., & Pras, A. (2009). Hidden Markov Model Modeling of SSH Brute-Force Attacks. In C. Bartolini & L. P. Gaspary (Eds.), *Integrated Management of Systems, Services, Processes and People in IT* (pp. 164–176). Springer. https://doi.org/10.1007/978-3-642-04989-7_13
- Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., & Stiller, B. (2010). An Overview of IP Flow-Based Intrusion Detection. *IEEE Communications Surveys & Tutorials*, 12(3), 343–356. <https://doi.org/10.1109/SURV.2010.032210.00054>
- Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Secure shell (ssh) traffic analysis with flow based features using shallow and deep networks. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2026–2032. <https://doi.org/10.1109/ICACCI.2017.8126143>