Mehran University Research Journal of Engineering and Technology

https://doi.org/10.22581/muet1982.2301.14

2023, 42(1) 153-164

Network intrusion detection system using an optimized machine learning algorithm

Abdulatif Alabdulatif [a], Syed Sajjad Hussain Rizvi [b,*]

[a] *Computer Department, College of Science and Arts in Ar Rass, Qassim University, Ar Rass Saudi Arabia*

[b] *Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Karachi Pakistan*

[*] Corresponding author: Syed Sajjad Hussain Rizvi Email: dr.sajjad@szabist.edu.pk

Received: 19 September 2022, Accepted: 15 December 2022, Published: 01 January 2023

KEYWORDS

Network Intrusion Detection

Machine Learning

Hyper-Parameter Optimization

Kitsune

Cyber Security

Communication Network

ABSTRACT

The rapid growth of the data-communications network for real-world commercial applications requires security and robustness. Network intrusion is one of the most prominent network attacks. Moreover, the variants of network intrusion have also been extensively reported in the literature. Network Intrusion Detection Systems (NIDS) have already been devised and proposed in the literature to handle this issue. In the recent literature, Kitsune, NIDS, and its dataset have received approx. 500 citations so far in 2019. But, still, the comprehensive parametric evaluation of this dataset using a machine learning algorithm was missing in the literature that could submit the best algorithm for network intrusion attack detection and classification in Kitsune. In this connection, two previous studies were reported to investigate the best machine algorithm (these two studies were reported by us). Through these studies, it was concluded that the Tree algorithm and its variants are best suited to detect and classify all eight types of network attacks available in the Kitsune dataset. In this study, the hyper-parameter optimization of the optimized Tree algorithm is presented for all eight types of network attack. In this study, the optimizer functions Bayesian, Grid Search, and Random Search were chosen. The performance has been ranked based on training and testing accuracy, training and testing cost, and prediction speed for each optimizer. This study will submit the best point hyper-parameter for the respective epoch against each optimizer.

1. Introduction

Since the last decade, the NIDS has gained major popularity in the domain of cyber security [1]. It is because of a multi-folded reason which includes, but is not limited to, integration of business applications with communication networks; global integration of businesses; online businesses and ventures; remote working; etc. [2]. The performance, efficiency, and robustness of the classical NIDS were significantly uplifted by integrating the NIDS with artificial intelligence [3]. Although in the recent literature, the researchers have proposed various Deep Learning and Machine Learning based solutions to make NIDS effective and competent in identifying malevolent attacks. However, the rapid increase in the network traffic and the emerging safety threats has posed by challenges for NIDS system for the detection of malevolent intrusions efficiently. In recent literature, significant work has been reported in the common domain of NIDS and artificial intelligence.

Multi-stage optimization of machine learning algorithm for NIDS [4], BAT modeling for NIDS [5], Boosting algorithm for NIDS, 1D CNN based NIDS [6], stacking ensemble for NIDS [7], ensemble-based NIDS [8] etc. However, all these existing methods are found to be deficient for optimal machine learning perimeters. Whereas the proposed algorithm is trained for real time NIDS and have been tuned for optimal machine learning perimeter. For robust and intelligent NIDS, machine learning and deep learning-based algorithms require well-structured and domain-oriented datasets. In this connection, many researchers and research groups have submitted the dataset in the domain of NIDS. The well-known NIDS datasets are LITNET-2020 [9], Netflow [10], CIDDS-001 [11], UNSW-NB15 [12], Kitsune [13], and many more are in the row.

Each dataset has its own objectives and respective constraints. This study is mainly focused on the Kitsune dataset. In the literature, a comprehensive comparative study on the utility of machine learning algorithms was found to be deficient. In one of our previous studies, a comprehensive parametric performance evaluation of a machine learning algorithm to detect the Mirai Botnet attack was presented [14].

In this study, a larger variant of machine learning algorithms was employed on the Kitsune dataset for Mirai Botnet attack detection and classification. The algorithms used in this study are Fine Tree, Medium Tree, Coarse Tree, Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, Coarse Gaussian SVM, Boosted Tree, Bagged Tree, Subspace Discriminant, RUSBoosted Tree, Logistic Regression, and Gaussian Naïve Bayes.

The performance of each algorithm was measured as the function of the confusion matrix, the True Positive (TPR), False Negative Rate (FNR), Accuracy of No-Attack, Accuracy of Attack, Net Accuracy, Test Accuracy, Misclassification Cost, Prediction Speed, and Train Time. This study reveals that the variants of the Tree algorithm have submitted comparatively better performance parameters as compared to other variants of machine learning algorithms.

This finding has established the hypothesis that the same variants of tree algorithms may also perform relatively better for other types of network attacks available in the Kitsune dataset. In 2021, another study was reported to validate the above hypothesis [15]. In this study, variants of tree algorithms were employed on the all-network attacks of Kitsune.

The attacks include Active Wiretap, ARP MitM, Fuzzing, OS Scan, SSDP Flood, SSL Renegotiation, SYN Dos, and Video Injection. The performance was also measured as the function of the training accuracy, test accuracy, misclassification cost, prediction speed, and train time. In the same year, another study was conducted in which the research proposed a hybrid soft computing-based network for intelligent energy and efficiency management system.

In this work, the authors proposed a model based on the evolutionary neuro-fuzzy approach that could predict the energy demand as an objective function and optimize the energy within the given constraints [16].

Another work in the same year was presented a comprehensive review about the strengths and limitations of existing data driven approaches of for energy efficiency management and optimization. [17].

In 2022, study has presented a comprehensive and exhaustive empirical evaluation of machine learning algorithm for energy demand prediction on the SEIL dataset. In the study the simulation results established the findings that Bagged Trees is most effective algorithm for the said application and Medium Trees is the most efficient one [18].

The finding of this research paper validates our hypothesis that the variants of the Trees algorithm also perform to acceptable accuracy for the other types of network attacks given in the Kitsune dataset. The validation of the hypothesis has motivated to further extend the scope of the work towards the optimization of the best variants for Kitsune network attack detection and classification. In this study, the optimization of the variants of the tree algorithm for NID attack detection and classification is presented.

The three fundamental optimizers were used in this study, namely, Bayesian, Grid search, and Random search. Moreover, the results are compared with our previous work. The scope of this study is to evaluate different optimizers on the variants of the tree algorithm for Kitsune network attack classification (all network attacks reported in Kitsune). The optimized variant will submit the best point optimizer for each network attack training.

2. Methodology

This section will describe the approach that was employed in this investigation. The optimization of tree algorithm variants for NID attack classification and detection is presented in this study. Fundamental

optimizers include Bayesian, grid, and random searches. To determine the optimal hyper-parameters, the approach also takes into account previous findings.

2.1 Grid Search

Because it exhaustively explores hyperparameter sets in a given range, grid search is the most often used approach for hyperparameter optimization. Users should have prior knowledge of these hyperparameters while picking all candidates. Grid search can be used to look for numerous hyperparameters at once. Grid search is the most essential and important search algorithm, producing the utmost correct predictions and allowing users to always find the finest and optimum combination. Grid search has the advantage of being mathematically simple and clear and capable of running numerous parameters concurrently. However, the major drawback is that the overall search time increases as the number of hyper-parameters to be searched increases.

2.2 Random Search

Random search is a significant advancement over grid search. The search process is repeated until the required accuracy or the predetermined number of iterations is obtained. Random search is identical to grid search, however it has been found to provide superior results due to the following two advantages: The first is that a random search works better, particularly when some hyperparameters are not distributed uniformly. Random searches locate the ideal configuration in this search pattern more efficiently. Second, the quality of the random searches will be significant. The more time spent, the more probable a superior collection of hyperparameters will be discovered.

2.3 Bayesian Optimization

Bayesian optimization is designed to detect an optimal solution that maximize or minimize the function value by presuming an unknown objective function that gets an input.

With these techniques, the data set undergoes pre-processing, then experimental setup. The data set is then trained and then checked for competency and then it is optimized. Whereas the performance has been ranked on the basis of training and testing accuracy, training and testing cost, and prediction speed for each optimizer. This study will submit the best point hyper-parameter for the respective epoch against each optimizer.

3. Simulation Results and Analysis

This section will present a comprehensive analysis of the simulation results. Table 1 presents a parametric performance evaluation of the optimized tree algorithm with the three optimizers, namely, Bayesian, Grid search, and Random search.

It has been validated that all of the optimizers for tree algorithms have reported a 100% training and testing accuracy. However, the un-optimized tree algorithm, reported in our previous study, has reported a marginally lower testing accuracy. Therefore, a marginal improvement in the test accuracy has been observed for the optimized tree. In Table 1, the best training time illustrates the minimum training time for the respective algorithm in our previous study of un-optimized tree algorithms for network attack detection. If the best training is compared with the training time of the optimized tree, a significant difference can be witnessed.

This is because a series of values was used to find the best point optimizer in the optimized training; thus, multiple trainings for a series of values take a long time. But, once the best point hyper-parameters are achieved, then it costs less time. In this study, no significant improvement can be observed in the predicted speed between an optimized tree and an un-optimized tree. This finding refutes the hypothesis that the optimized tree algorithm will produce significantly better predictions in network attack detection. Therefore, the hypothesis can be marked as rejected. This experimental finding has furnished the conclusion that no significant improvement in perdition speed can be established with the optimized variants of the tree algorithm for Kitsune NIDS.

To complement the discussion, the graphical illustrations of the tabulated findings in Table 1 are illustrated in Fig. 1 to 4 for training cost, prediction speed, testing cost, and training time, respectively. It can be inferred from these figures that a proportional change has occurred in the training cost, testing cost, training time, and prediction speed for both the optimized and un-optimized tree algorithms in Kitsune NIDS. After careful analysis of these figures, it has been observed that the grid search algorithm has submitted relatively better performance in terms of training cost, testing cost, training time, testing time, and prediction speed as compared to the other candidates.

Table 1

Performance evaluation of optimize tree algorithm with optimizers

		Training						Testing	
Network Attack	Optimizer	Accuracy	Total Cost	Prediction Speed (ops/sec)	Training Time (sec)	Best Training time (sec)	Best Prediction Speed (ops/sec)	Accuracy	Total Cost
Active Wiretap	Bayesian	100	1	1000000	1153.7	461	1100000	100	7
	Grid search	100	1	1000000	747.07			100	5
	Random search	100	1	900000	1122.8			100	5
ARP MitM	Bayesian	100	2	610000	1255.6	578	1400000	100	7
	Grid search	100	2	1100000	831.86			100	7
	Random search	100	2	1100000	1220.8			100	7
Fuzzing	Bayesian	100	4	910000	891.15	383	1200000	100	6
	Grid search	100	2	1000000	600.12			100	2
	Random search	100	2	1000000	871.66			100	2
OS Scan	Bayesian	100	1	810000	559.41	194	1000000	100	8
	Grid search	100	1	700000	369.7			100	8
	Random search	100	1	98000	565.32			100	8
SSDP Flood	Bayesian	100	1	860000	1991.6	1232	1200000	100	1
	Grid search	100	1	800000	1322.1			100	1
	Random search	100	1	810000	2082.4			100	2
SSL Renegotiation	Bayesian	100	7	690000	2013.7	616	10000000	100	16
	Grid search	100	7	700000	1359.8			100	16
	Random search	100	7	450000	2173.7			100	16
SYN DOS	Bayesian	100	14	790000	1611.3	548	740000	100	6
	Grid search	100	14	700000	1053.8			100	6
	Random search	100	14	670000	1204.2			100	6
Video Injection	Bayesian	100	2	900000	1227.4	639	1300000	100	7
	Grid search	100	1	1100000	826.45			100	6
	Random search	100	1	1000000	1132.4			100	6

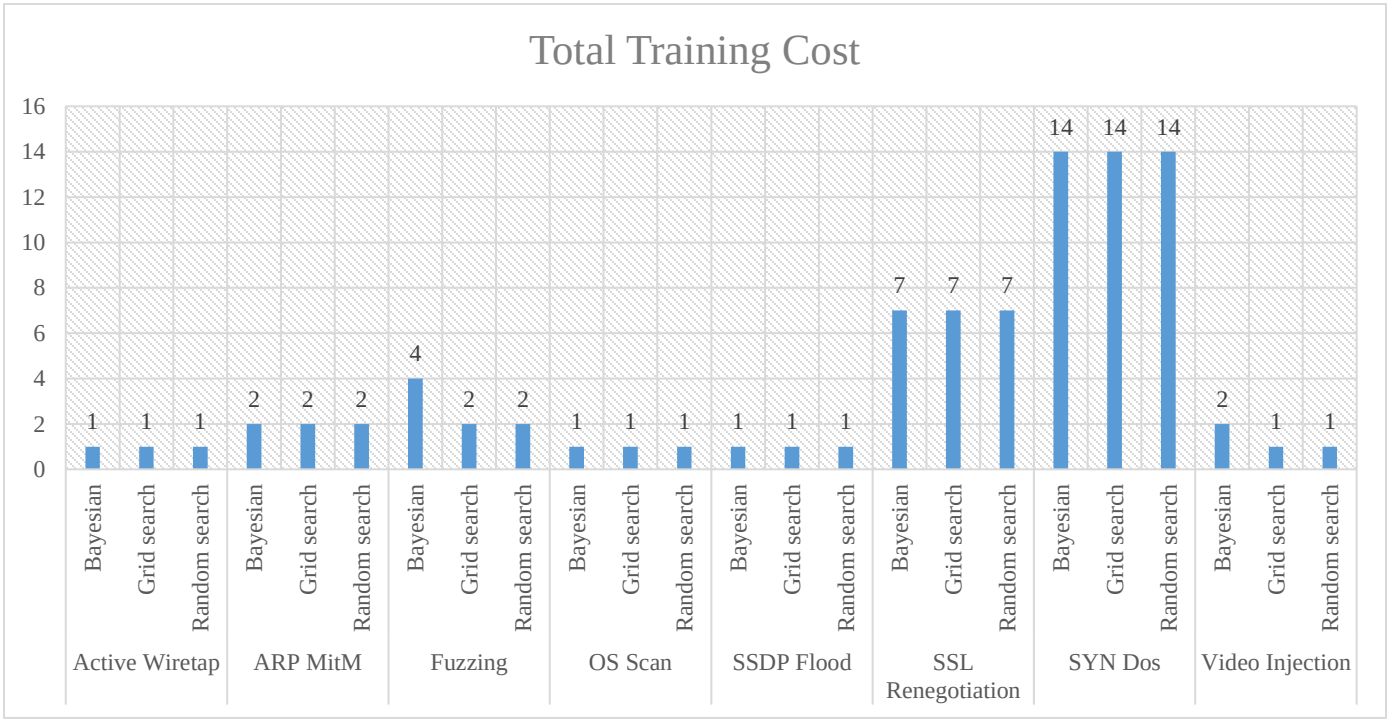

Fig. 1. Total training cost

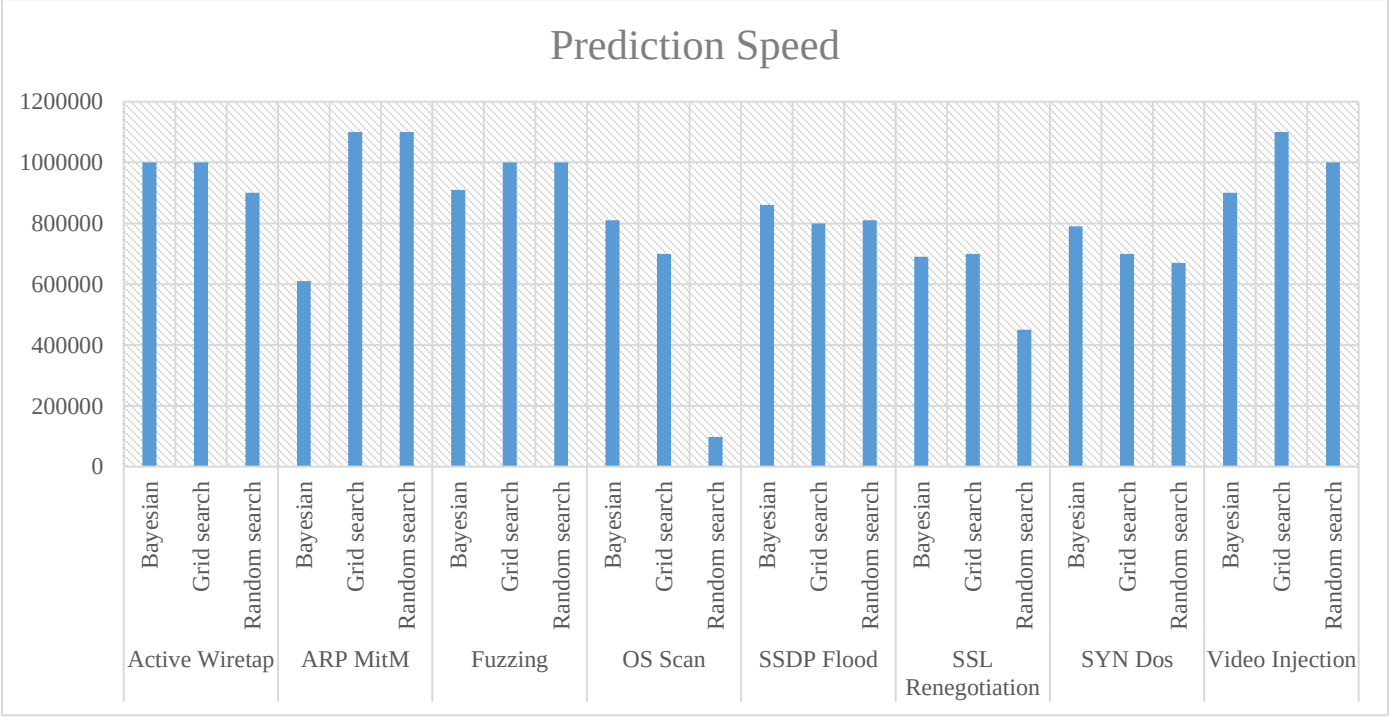

Fig. 2. Prediction speed

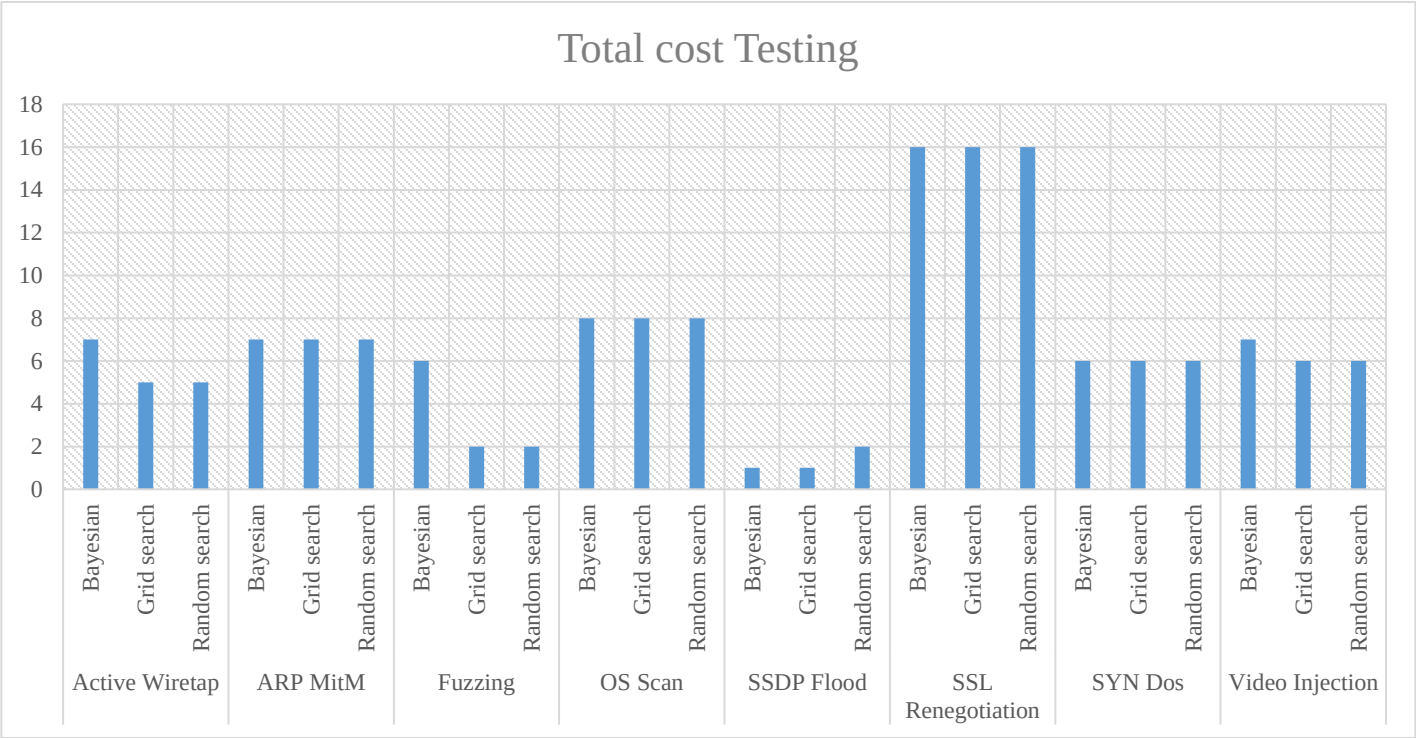

Fig. 3. Total cost testing

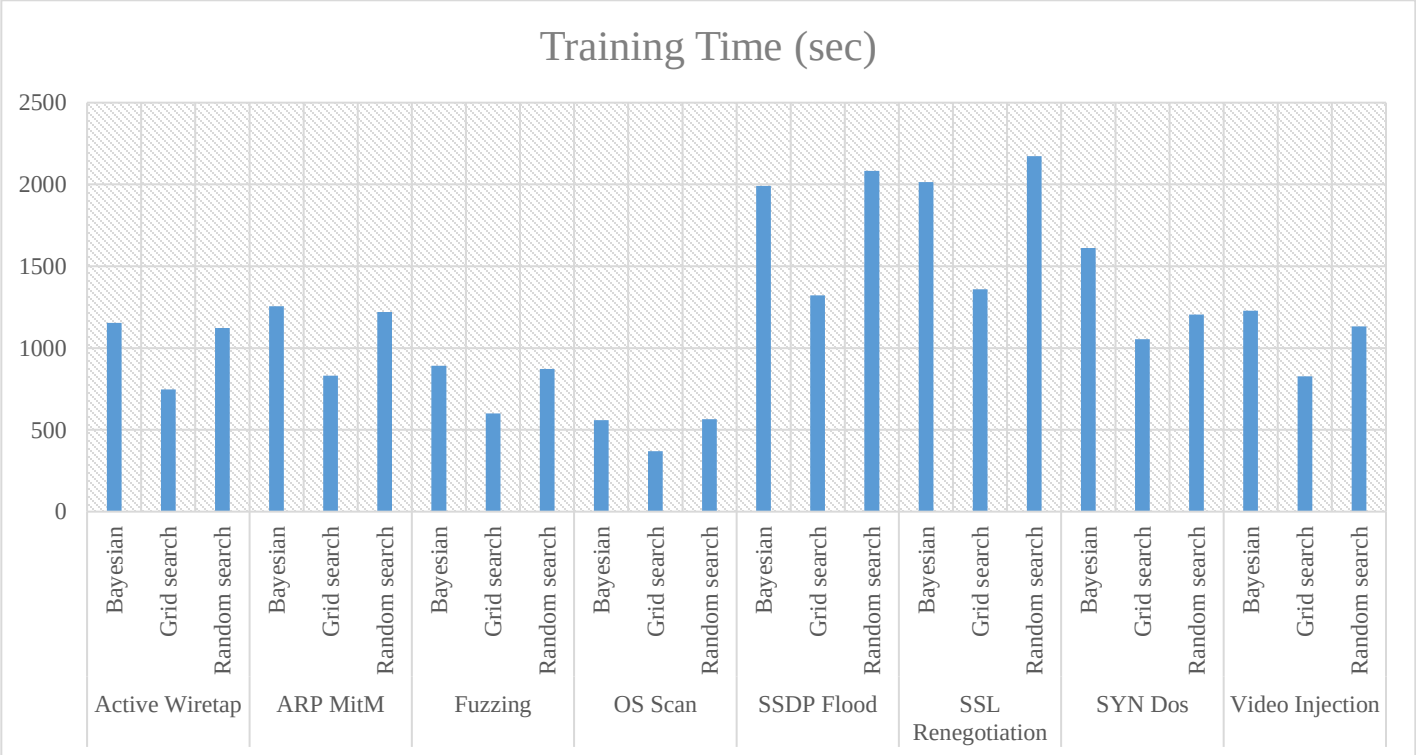

Fig. 4. Training time (sec)

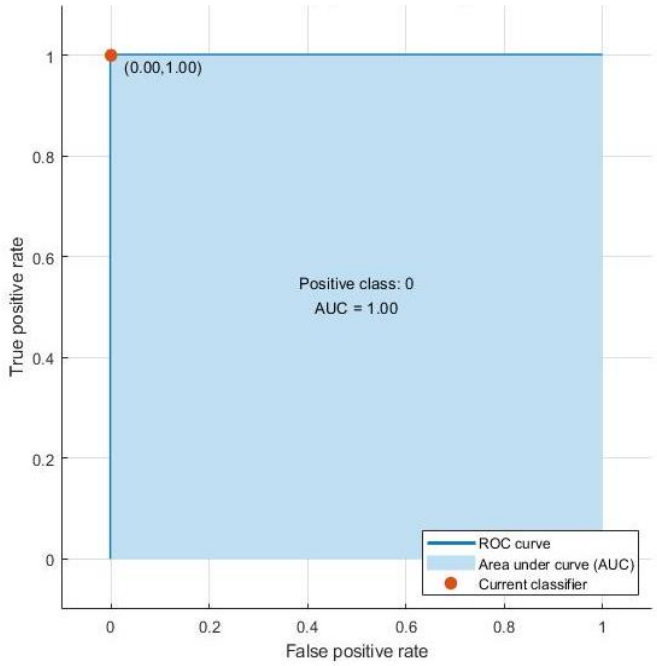

Fig. 5. ROC curve of active wiretap

Fig. 5 shows the ROC curve for the Active wiretap network attack detection. Since the training and testing accuracy is reported to be 100% therefore, the AUC is reported to be 1. The similar ROC was reported by the other network attack, therefore only the RoC (receiver operating characteristic curve) for Active Wiretap has been presented for the ready reference. In the subsequent part of this section the minimum classification error with respect the epochs are presented as the graphical illustration for all three optimizers under considerations. In the subsequent graphs the light blue line represents the estimated minimum classification error, the dark blue line represents the observed minimum classification error, the red pointer shows the best point hyper-parameters as the function of the epoch, and yellow marker shown the minimum error hyper-parameter.

Fig. 6 to 8 shown the minimum classification error for Bayesian, grid, and random search optimizers for Active Wiretap network attack detection. It has been observed that the Random search optimizer is found to be relatively good candidate due to minimum classification error and the best point epoch hype parameters is reported at 9th iteration. However, the other optimizer has reported a relatively larger epoch as the best point hyper-parameter. Likewise, Fig. 9 to 11 shown the minimum classification error for Bayesian, grid, and random search optimizers for ARP MitM network attack detection, respectively. The analysis of these simulation results reveals the fact that the grid search optimization submits the minimum epoch required to achieve the minimum classification error. The grid search algorithm has reported approximately three times lesser epochs required for minimum classification error.

The similar response has also been witnessed in the Fig 12 to 14 for fuzzing network attack, where the grid search optimizer also won the race of optimization with significantly lesser number of epochs as the best optimizer parameters in order to achieve the minimum classification error. The analysis of training of machine learning algorithm for the OS scan network attack also complies to the finding that the grid search optimizer is found to be the best optimizer for network training. The grid search optimization is tuned to perform the best training parameter at the epoch of 11 that is far lesser than the other variant of optimization in the group.

The minimum classification error shown in Fig. 18 to 20 also advocated to the grid search optimizer to be employed in the Kitsune for SSDP Flood attack detection. The grid search optimizer has best tuned the tree algorithm for the six epochs to have the minimum classification error. However, the other optimizer like Bayesian and random search has submitted the relative high value of epoch tuned to report the minimum classification error. The minimum classification error of the training of Kitsune for SSL Renegotiation is illustrated in Fig. 21 to 23 for Bayesian, grid search, and random search optimizer. It has been figured the in SSL Renegotiation the grid search optimizer to perform better over the other optimizer in the group for minimum classification with only one epoch.

The minimum classification error for the training of Kitsune for SYN DOS and Video Injection are illustrated in Fig. 24 to 29 for Bayesian, grid search and random search algorithm respectively. The analysis of these simulation results advocated for the random search optimizer. It can be inferred from these simulation results that the recommendation should be placed to use the random search algorithm to achieve the best point hyper-parameter. In the light of the above findings, it is concluded that the tree algorithm is best suited for Kitsune network attack detection and the grid search optimizer is the corresponding best optimizer for best point hyper-parameter.

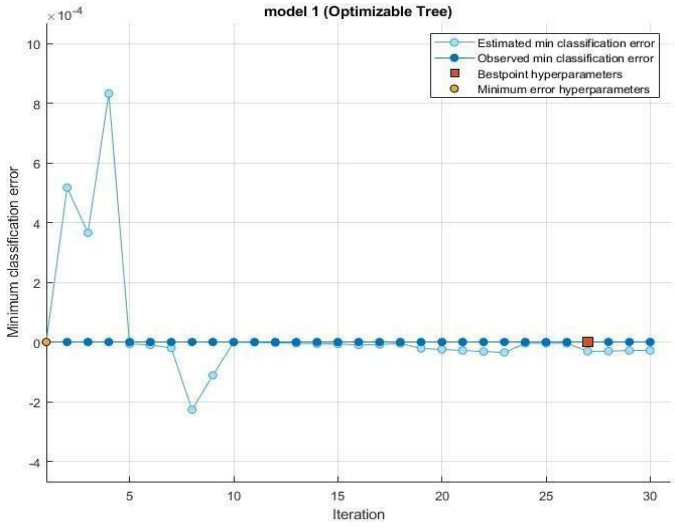

Fig. 6. Active wiretap minimum classification error of bayesian based optimized tree

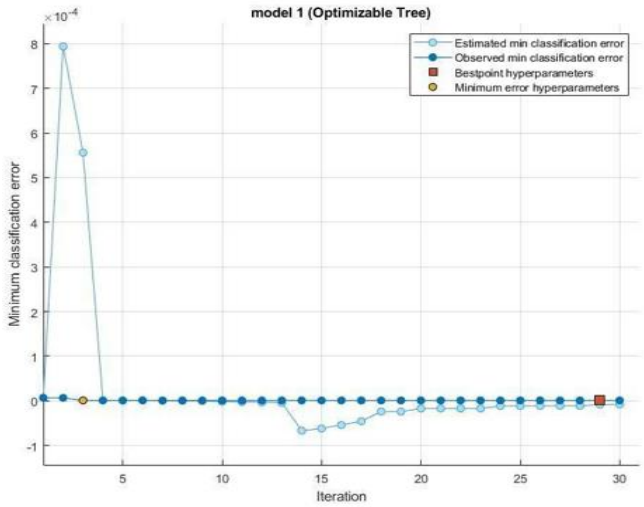

Fig. 9. ARP MitM minimum classification error of bayesian based optimized tree

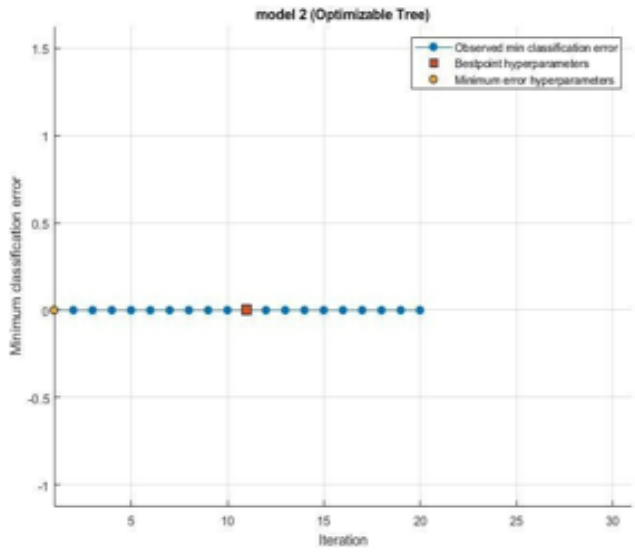

Fig. 7. Active wiretap minimum classification error of grid search based optimized tree

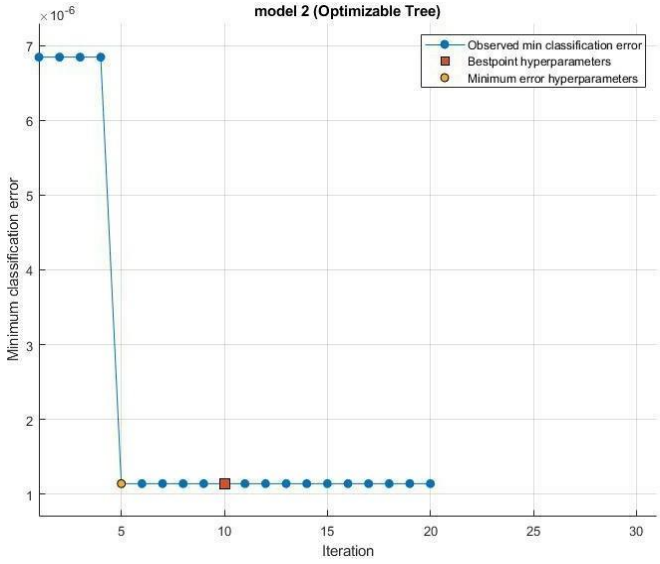

Fig. 10. ARP MitM minimum classification error of grid search based optimized tree

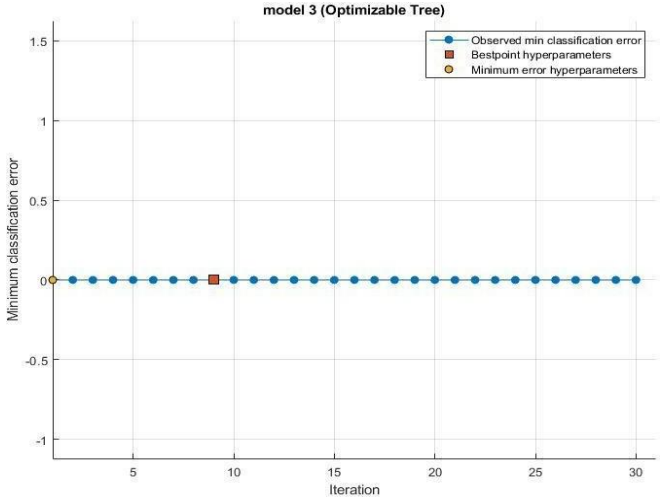

Fig. 8. Active wiretap minimum classification error of random search based optimized tree

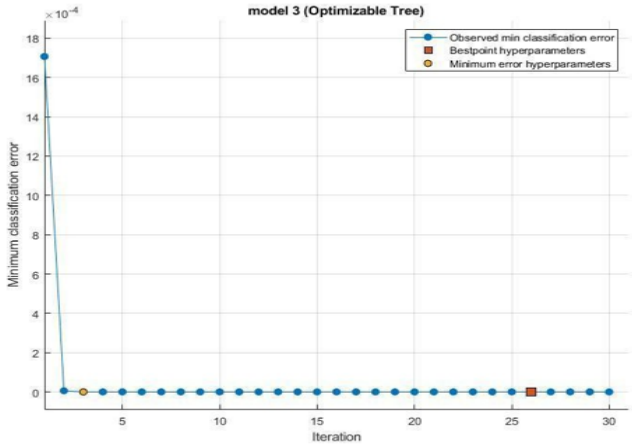

Fig. 11. ARP MitM minimum classification error of random search based optimized tree

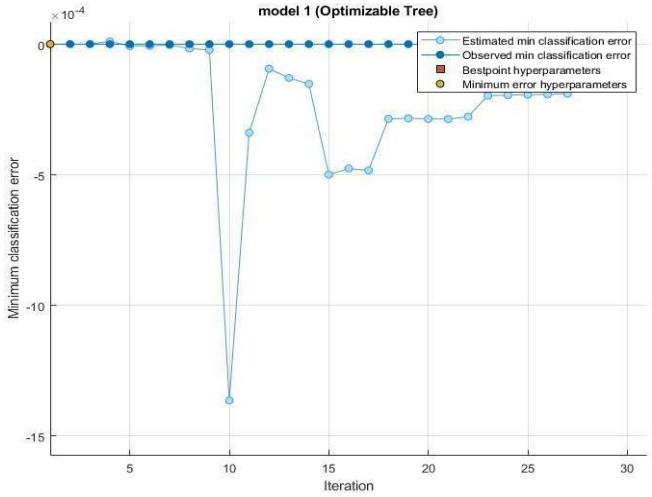

Fig. 12. Fuzzing minimum classification error of bayesian based optimized tree

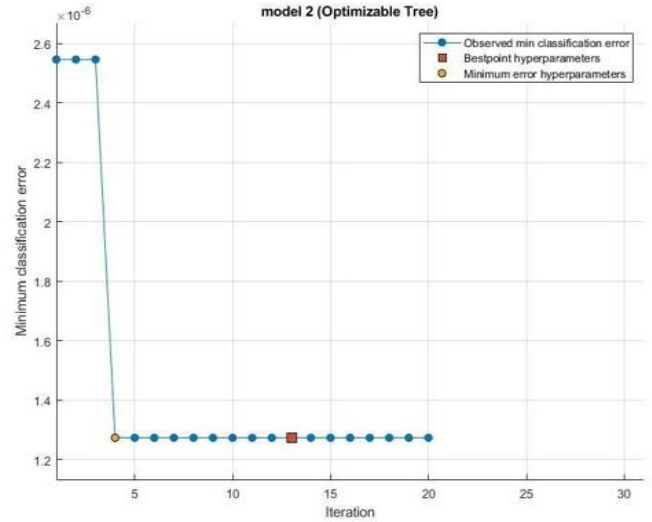

Fig. 13. Fuzzing minimum classification error of grid search based optimized tree

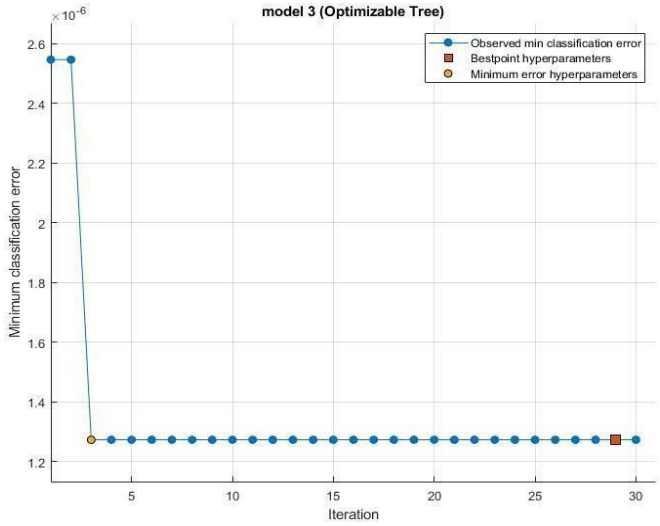

Fig. 14. Fuzzing minimum classification error of random search based optimized tree

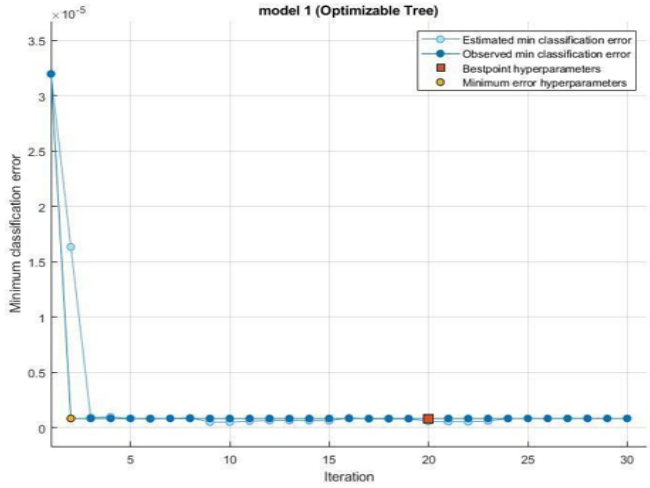

Fig. 15. OS scan minimum classification error of bayesian based optimized tree

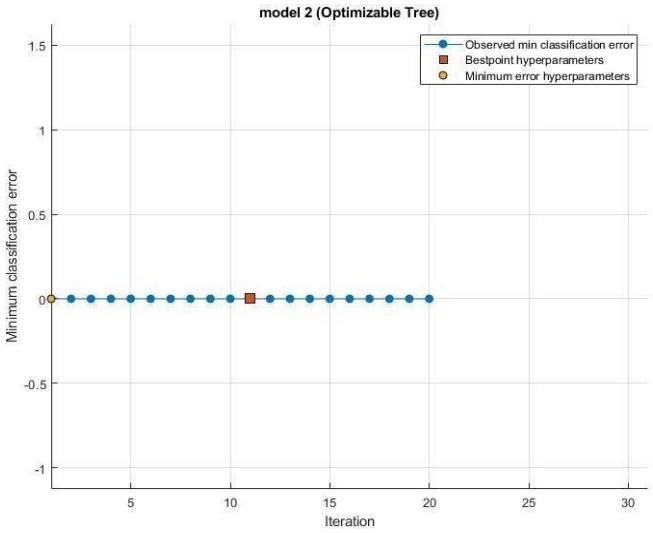

Fig. 16. OS scan minimum classification error of grid search based optimized tree

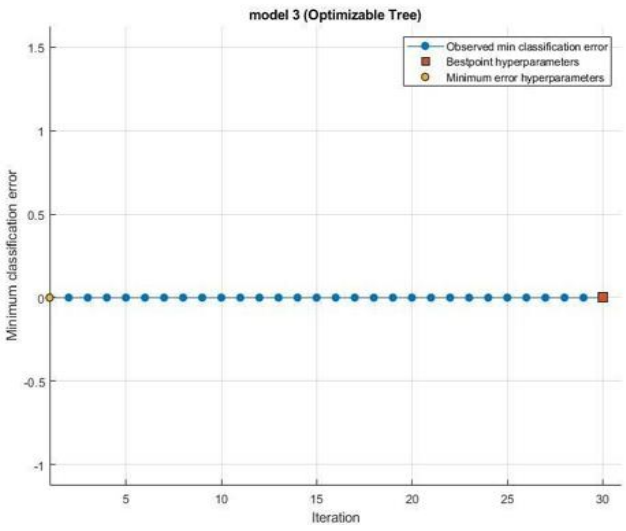

Fig. 17. OS scan minimum classification error of random search based optimized tree

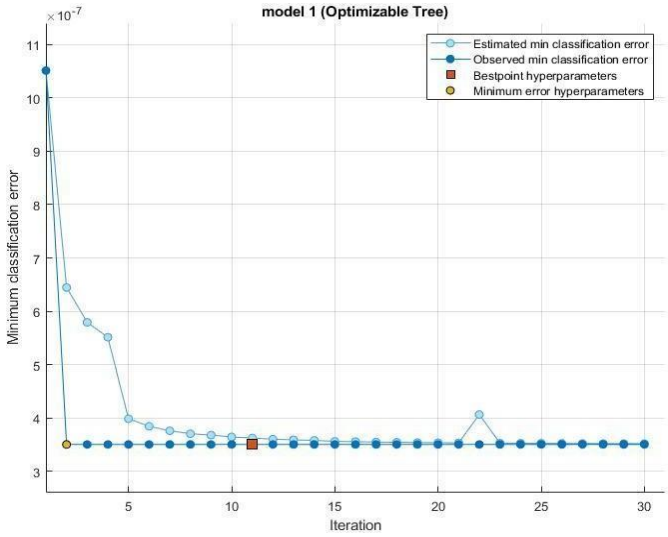

Fig. 18. SSDP flood minimum classification error of bayesian based optimized tree

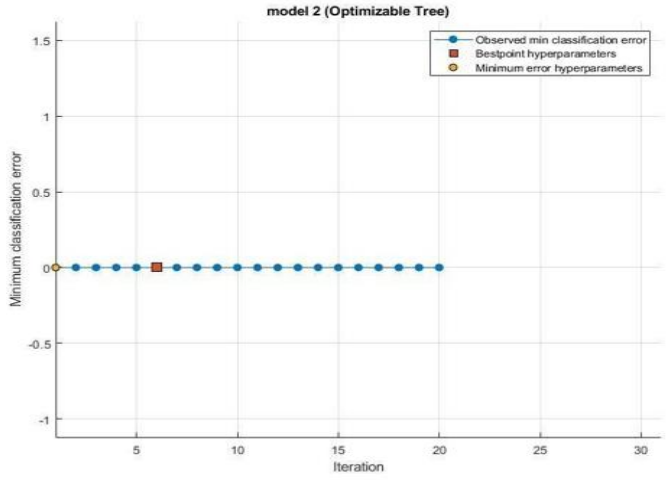

Fig. 19. SSDP flood minimum classification error of grid search based optimized tree

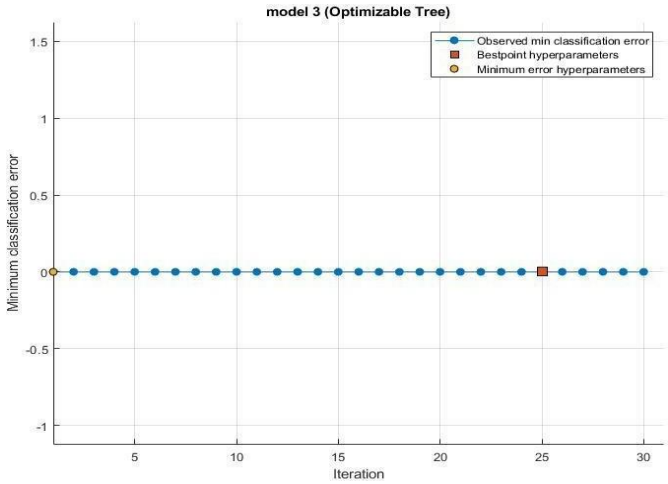

Fig. 20. SSDP flood minimum classification error of random search based optimized tree

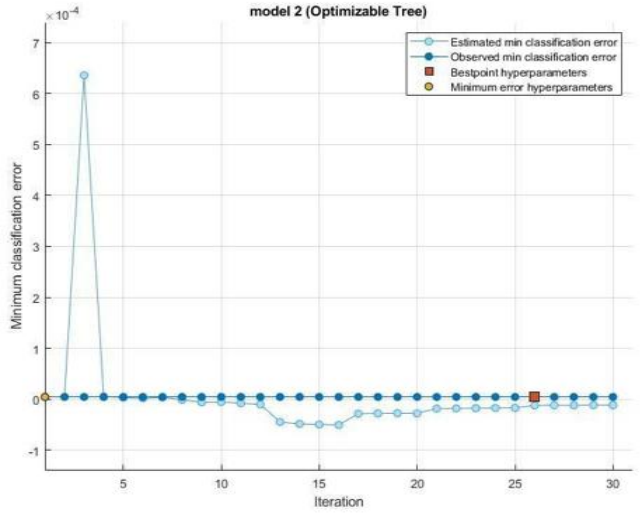

Fig. 21. SSL renegotiation minimum classification error of bayesian based optimized tree

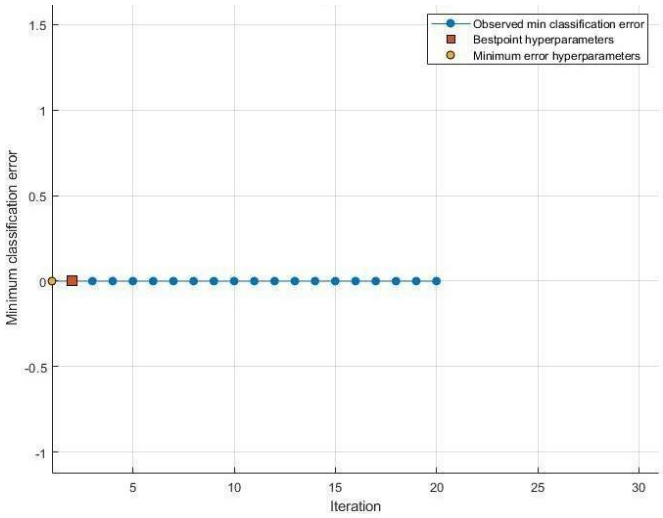

Fig. 22. SSL renegotiation minimum classification error of grid search based optimized tree

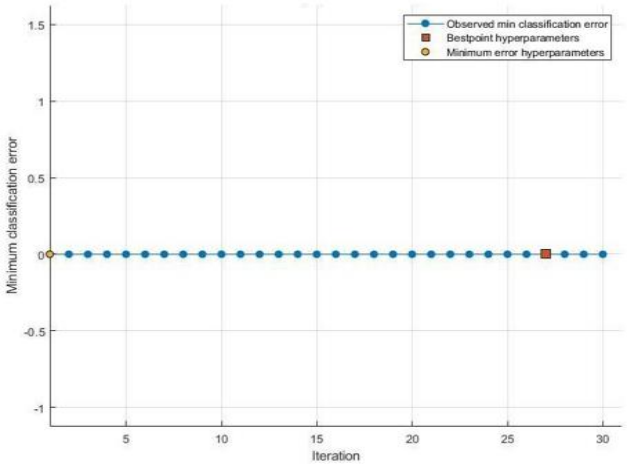

Fig. 23. SSL renegotiation minimum classification error of random search based optimized tree

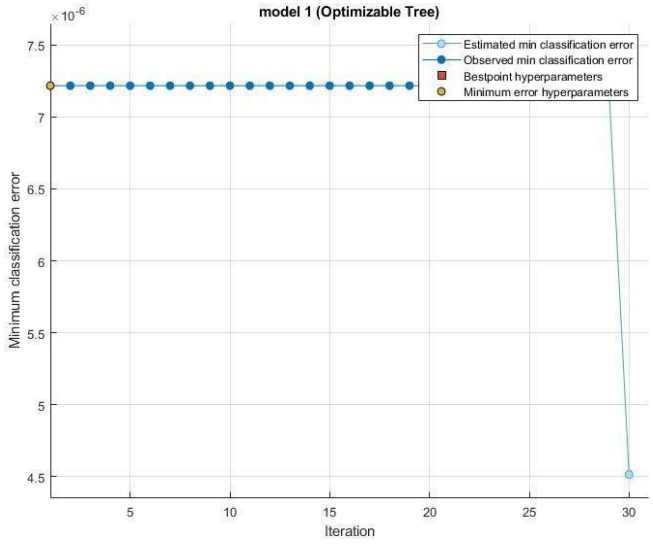

Fig. 24. SYN DOS minimum classification error of bayesian based optimized tree

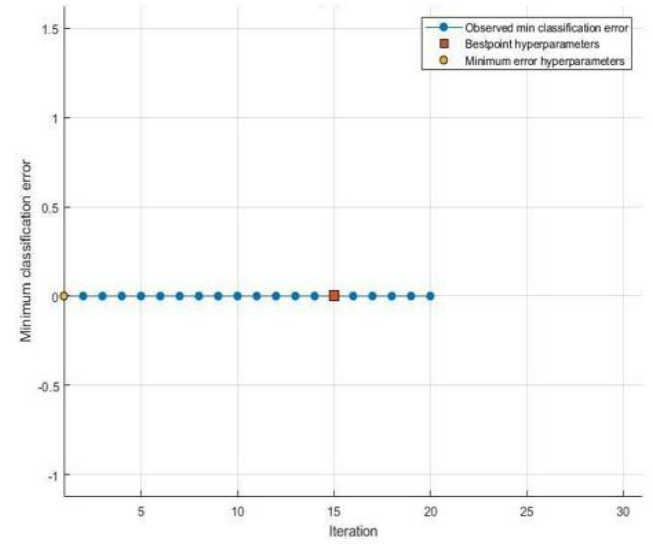

Fig. 25. SYN DOS minimum classification error of grid search based optimized tree

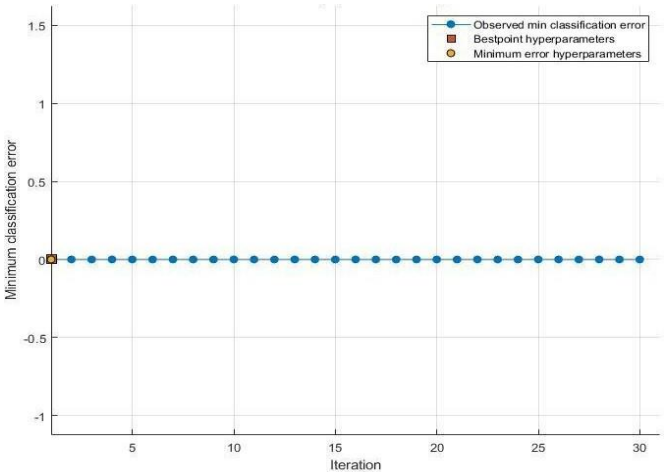

Fig. 26. SYN DOS minimum classification error of random search based optimized tree

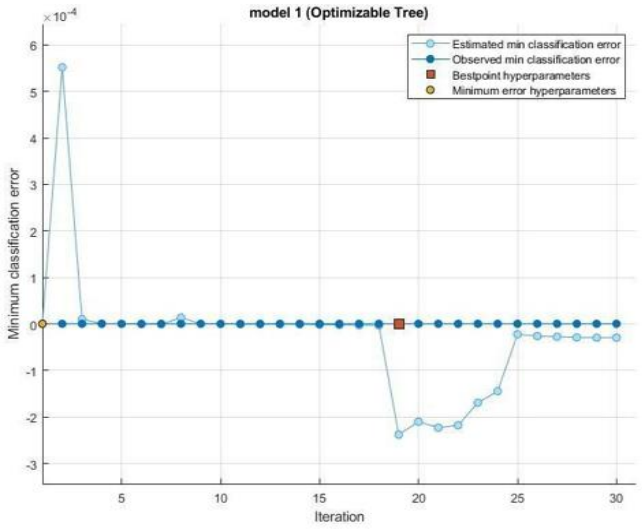

Fig. 27. Video injection minimum classification error of bayesian based optimized tree

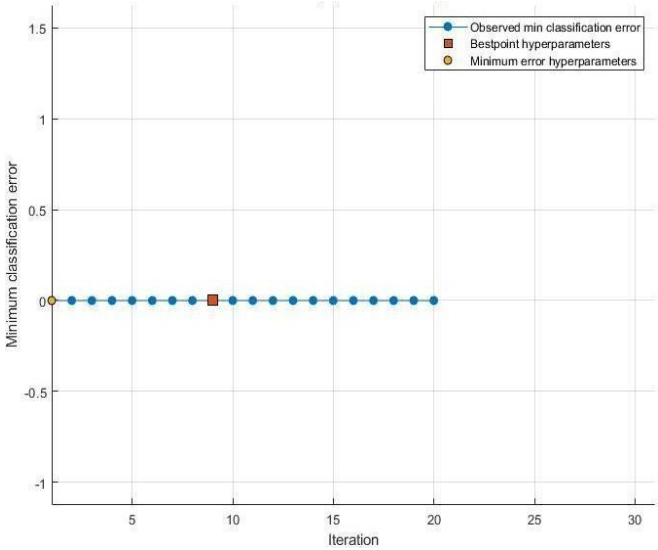

Fig. 28. Video injection minimum classification error of grid search based optimized tree

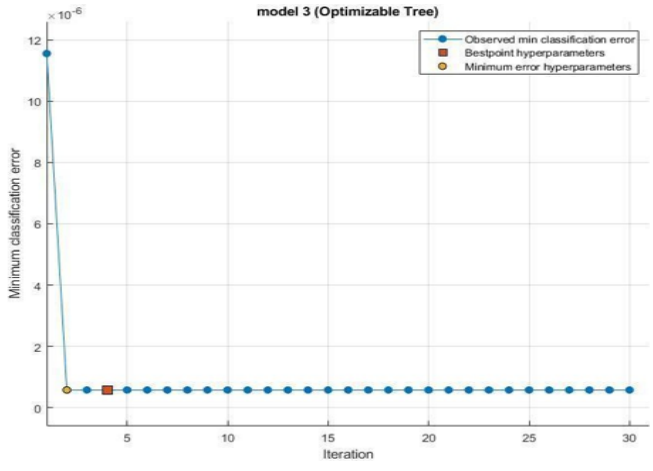

Fig. 29. Video injection minimum classification error of random search based optimized tree

4. Conclusion

This study has presented the recommendation of an optimized method for Kitsune for machine learning-based network attack detection and classification. In our previous study, it was shown that the Tree algorithm is found to be the best candidate for Mirai Botnet Attack detection and classification. In continuation of this study, it has been proved that the Tree algorithm is best suited for all other network attacks defined in Kitsune. Finally, in this study, the grid search optimizer is recommended with the Tree algorithm for network attach detection and classification in Kitsune NIDS. The rationale for the said claim was justified with the simulation results and analysis.

5. References

[1] J. Kaiyuan et al. "Network intrusion detection combined hybrid sampling with deep hierarchical network", IEEE Accessing 8 (2020): 32464-32476.

[2] H. S. Lallie et al. "Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic", Computers and Security 105 (2021): 102248.

[3] Z. Zhang et al. "Artificial intelligence in cyber security: research advances, challenges, and opportunities", Artificial Intelligence Review (2021): 1-25.

[4] M. N. Injadat et al., "Multi-stage optimized machine learning framework for network intrusion detection", IEEE Transactions on Network and Service Management 18.2 (2020): 1803-1816.

[5] T. Su et al., "BAT: deep learning methods on network intrusion detection using NSL-KDD dataset", IEEE Accessing 8 (2020): 29575-29585.

[6] A. Shahraki, M. Abbasi, and Ø. Haugen., "Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost", Engineering Applications of Artificial Intelligence 94 (2020): 103770.

[7] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "A stacking ensemble for network intrusion detection using heterogeneous datasets", Security and Communication Networks 2020.

[8] J. Zhang, F. Li, and F. Ye, "An ensemble-based network intrusion detection scheme with bayesian deep learning", International Conference on Communications, IEEE, 2020.

[9] R. Damasevicius et al., "LITNET-2020: An annotated real-world network flow dataset for network intrusion detection", Electronics 9.5 (2020): 800.

[10] M. Sarhan et al., "Netflow datasets for machine learning-based network intrusion detection systems", Big Data Technologies and Applications. Springer, Cham, pp. 117-135, 2020.

[11] A. Verma and V. Ranga, "Statistical analysis of CIDDS-001 dataset for network intrusion detection systems using distance-based machine learning", Procedia Computer Science 125 (2018): 709-716.

[12] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT", Procedia Computer Science 167 (2020): 1561-1573.

[13] Y. Mirsky et al., "Kitsune: an ensemble of autoencoders for online network intrusion detection", arXiv preprint arXiv:1802.09089 (2018).

[14] A. Abdullah, S. S. H. Rizvi, and M. A. Hashmani. "Optimal Machine Learning Models for Kitsune to Detect Mirai Botnet Malware Attack." Journal of Hunan University Natural Sciences 48.6 (2021).

[15] A. Abdullah, S. S. H. Rizvi, "Machine Learning Approach for Improvement in Kitsune NID", Intelligent Automation and Soft Computing 32.2 (2022): 827-840.

[16] S. Akhtar, Z. B. Sujod, S. S. H. Rizvi, "A hybrid soft computing framework for electrical energy optimization", 6th International Multi-Topic ICT Conference 2021.

[17] S. Akhtar, Z. B. Sujod, S. S. H. Rizvi, "A Novel Deep Learning Architecture for Data-Driven Energy Efficiency Management - Systematic Survey", 7th International Conference on Engineering and Emerging Technologies", October 2021, Istanbul, Turkey.

[18] S. Akhtar, M. Z. B. Sujod, S. S. H. Rizvi, "An intelligent data-driven approach for electrical energy efficiency management", Energies MDPI.

