Machine learning techniques for accurate classification and detection of intrusions in computer network

Mutyalaiah Paricherla[1], Mahyudin Ritonga[2], Sandip R. Shinde[3], Smita M. Chaudhari[4], Rahmat Linur[5], Abhishek Raghuvanshi[6]

[1]Department of Computer Science and Engineering, NBKR Institute of Science and Technology, Vidyanagar, India
[2]Department of Arabic Language and Education, Faculty of Islamic Studies, Universitas Muhammadiyah Sumatera Barat, Padang, Indonesia
[3]Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India
[4]Department of Computer Engineering, Marathwada Mitra Mandal's College of Engineering, Pune, India
[5]Department of Arabic Language and Literature, Faculty of Ushuluddin and Dakwa, College of Islamic state Mandailing Natal, Mandailing Natal, Indonesia
[6]Department of Computer Science and Engineering, Mahakal Institute of Technology, Ujjain, India

Article Info

Article history:

Received Sep 5, 2022
Revised Oct 10, 2022
Accepted Dec 21, 2022

Keywords:

Accuracy
Adaboost
Ant colony optimization
F1 score
Gradient boosting
Intrusion detection
Prediction

ABSTRACT

An incursion into the computer network or system in issue occurs whenever there is an attempt made to circumvent the defences that are in place. Training and examination are the two basic components that make up the intrusion detection system (IDS) and each one may be analysed separately. During training, a number of distinct models are built, each of which is able to distinguish between normal and abnormal behaviours that are included within the dataset. This article proposes a combination of ant colony optimization (ACO) and the firefly approach for feature selection. The final outcome of giving careful thought to the selection of features will eventually result in greater accuracy of categorisation. When classifying various sorts of features, we make use of a wide variety of machine learning (ML) algorithms, including AdaBoost, gradient boost, and Bayesian network (BN), amongst others. The tests and assessments made use of data obtained from three distinct datasets, namely NSL-KDD, UNSW-NB15, and CICIDS 2017. The degree of performance of an individual may be broken down into its component parts, which include the F1 score, accuracy, precision, and recall. Gradient boost performs far better when it comes to recognising and classifying incursions.

Corresponding Author:

Abhishek Raghuvanshi
Department of Computer Engineering, Mahakal Institute of Technology
Madhya Pradesh, MIT Campus, Behind Air Strip, Dewas Rd, Ujjain, Madhya Pradesh 456001, India
Email: abhishek14482@gmail.com

1. INTRODUCTION

An intrusion occurs whenever there is an attempt to circumvent the protections that have been installed to secure a computer network or system [1]. Realizing that there was a problem with the system's security in the first place is the first step towards resolving the issue. Intrusion detection is both the method by which these attacks are detected and the term given to the system that is responsible for continuously checking network traffic for signals of prospective assaults [2]. Intrusion detection is the process by which these attacks are discovered.

The intrusion detection system (IDS) monitors every connection made to the network in order to identify any potential attacks. Intrusions are acts that are carried out with the intention of breaking into a computer system with the purpose of stealing information, disrupting service, or damaging hardware. When a user connects to your network from another part of the internet, there is a possibility that they may try to break in. This is something that a legitimate, authorised user could attempt to perform in order to increase the access permissions they have. One last consideration is the possibility that people with authorised access would misuse their powers. These undesired behaviours almost often put the network and/or the data it contains in jeopardy and use resources that may be put to greater use elsewhere [3], [4].

Training and testing are the two parts that make up the IDS [5]. During the training process, a variety of models are constructed that are able to differentiate between activities that are typical and actions that are not typical in dataset. The created models are graded according to their classification accuracy. It is common practice to choose an IDS according on how well it works on a testing dataset. In the context of teaching, to properly and effectively depict both typical and abnormal behavior [6]. Second, when it comes to modeling the behaviors, there are two ways available: supervised learning models and semi-supervised learning models. The difference between the two lies in whether or not the given training dataset contains the actual labels. A general model of intrusion detection is shown in Figure 1.

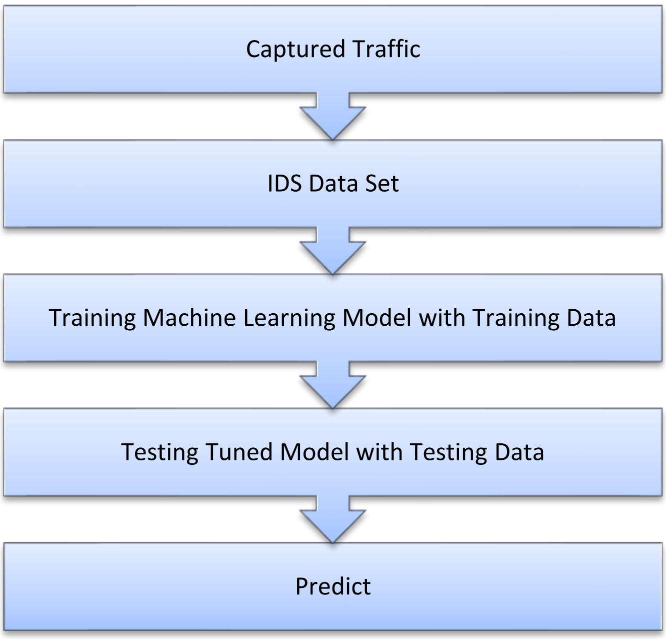

Captured Traffic
IDS Data Set
Training Machine Learning Model with Training Data
Testing Tuned Model with Testing Data
Predict

Figure 1. A block diagram for machine learning (ML) based IDS

This conversation will determine if an IDS can be added to the network's security architecture. IDS work alongside firewalls as the system's secondary line of defence [7]. The firewall prevents harmful Internet attacks from entering a business, and the IDS [8] alerts the system administrator of any security breaches when an attacker breaks the firewall and tries to access a trusted system. The traditional firewall filters inbound internet traffic, but hackers quickly find methods around it. Anyone can join to a private network's intranet using a modem [9]. A firewall couldn't predict such an entry point. An IDS examines network or computer data for signs of internal or external hostile activity.

Depending on the type of training data they employ, ML methods can be supervised, unsupervised, or semi-supervised. Unsupervised learning involves input samples without class labels [10]. The training dataset for supervised learning includes labelled samples. Professional data labelling may be expensive despite the vast amount of network and host data. Semi-supervised learning can use both tagged and unlabeled instances. Semi-supervised learning algorithms need a little quantity of labelled data to function, but they may use a big amount of unlabeled data. In the classification step, the created model or trained classifier assigns the test pattern to one of the pattern classes using the training-phase features. This follows training. ML-based IDS can rapidly identify threats while requiring less human labour. The system's learning capabilities enables this. This strategy is becoming increasingly significant for computer protection as the volume of network data IDS must review grows rapidly [11].

This article presents a framework to detect intrusion in network. This framework uses conjunction of ant colony optimization (ACO) and firefly method for selecting features from the intrusion data sets. This optimum feature selection results in increasing classification accuracy. ML techniques like-AdaBoost, gradient boost and Bayesian network (BN) are used to classify selected features. Performance is compared on the basis of accuracy, precision, recall, and F1 score.

2. LITERATURE SURVEY

The identification of insider threats was the primary focus of the work that Rane *et al.* [12] did for hybrid intrusion detection system (HIDS). For the sake of convenience, employees may choose to share their login credentials with one another; nevertheless, this may make the system more susceptible to infiltration in the event that the credentials are stolen or abused. System calls (SC) were collected as a result of user activity and the normal SC pattern of the user was taken into account while finishing up their profile. SC-level host IDS used forensic methods as well as other data mining techniques in order to identify any potential instances of internal attacks. In addition to the user's credentials, a forensic method was used in order to investigate the user's pattern of computer usage and compare it to the information included in the user profile. With a judgment rate threshold of 0.9, the system was able to reach an accuracy of 94%.

Research by Moon *et al.* [13] analysed the host user's activity pattern using advanced persistent threat assaults and 83 characteristics, each representing a personality trait. So, they may analyse the host user's conduct. The database was built using 8.7 million features from 4,000 malicious and benign programmes in a virtual machine (VM) environment. The system automatically calculates the incidence rates of all behaviours due to its architecture. Using the information, a C4.5 decision tree was used to classify each new occurrence as detrimental. The model's false positive and negative rates are 5.8%.

The research on identifying attacks using routing protocol for low-power and lossy networks (RPL). Malicious network activity might be found by converting RPL processes into finite state machines. Research by Le *et al.* [14] modelled RPL assaults using simulation traces. To learn more, we did this. We transformed the model into rules to govern network information flow. True positive rate reached 100%, whereas false positive rate ranged from 0% to 6.78%. 100% true positive rate. It utilised 6.3% more electricity than a normal RPL network.

Research by Ahmed *et al.* [15] found that the state transition approach could monitor protocol states and handle unexpected instructions. This was the study's conclusion. The researchers analysed state transition-based approaches for IDS and found that each had strengths and drawbacks. Tracing and validating protocol states is resource-intensive, and the system can't recognise ordinary assaults. Because a state transition diagram can only be produced for a succession of events, complex assaults will go unreported.

Research by Blowers and Williams [16] proposed density-based spatial application clustering with noise clustering. The system distinguished regular from irregular network packets. A 10% attack-to-no-attack pre-processing threshold was constructed, and features were picked using a correlation analysis of the knowledge discovery in databases (KDD) dataset. In ideal settings, the model identified possible threats 98% of the time. Research by Fawaz and Sanders [17] proposed KOBRA, a ML-based system to discover online abnormalities. It dissects application structure to understand behaviour. An ordered sequence of process events was created by using a number of kernel modules that can work together to examine the data. Time-stamped data are evaluated and correlated to get insight into programmed activities. Every piece of data was then given a 42-point anomaly score and compared to a threshold. The model identified the issue process.

Research by Muda *et al.* [18] presented an anomaly-based IDS to reduce false alarms while maintaining detection and accuracy. The suggested work is separated into two parts: first, k-means clustering is used to discriminate between attack and non-attack instances; next, the Naive Bayes classifier is used to categorise assaults into denial-of-service, remote code execution, user-to-user, and probing attacks. The proposal has two sections. The KDD cup 1999 data set was evaluated and showed a significant detection rate and 0.5% false alarm rate. ML main benefit is that it can adapt its own detection techniques.

ML delivers excellent detection rates and accuracy compared to other techniques. Clustering and a classifier offer a rapid reaction in real-time environments. The classifier's computation complexity decreases. Ensemble ML combines several less effective classifiers to get high-quality results [19]. Fusion-based ML reduces false warnings without affecting the system's ability to distinguish real hazards. Because a classifier's efficacy depends on assumptions and training, it has specific restrictions, just like any other approach. Comparatively, many resources are consumed. The grouping that was done was based on the hypothesis that smaller groups represent attacks that may be repelled by an attacker with more intelligence, while larger clusters represent typical data [20]. Establishing standard user profiles that may be used to identify an intruder requires a significant amount of effort and time. Because there is such a wide variety of classifiers, selecting the one that is best suited for the task may be a game of chance. The collection of features and the subsequent processing of those qualities by classifiers in a real-time scenario may be a time-consuming process.

3. METHOD

This section presents a framework (Figure 2) to detect intrusion in network. This framework uses conjunction of ACO and firefly method for selecting features from the intrusion data sets. This optimum feature selection results in increasing classification accuracy. ML techniques like-AdaBoost, gradient boost and BN are used to classify selected features. Performance is compared on the basis of accuracy, precision, recall and F1 score.

The ACO algorithm [21] and the firefly method [22] are often used in conjunction with one another in feature selection algorithms. The ACO algorithm performs the work for classification and analyses the continuity of features, while the firefly algorithm acts as a feature component of sub-band selection of intrusion related data. The creation of algorithms may be seen as an extension of ACO.

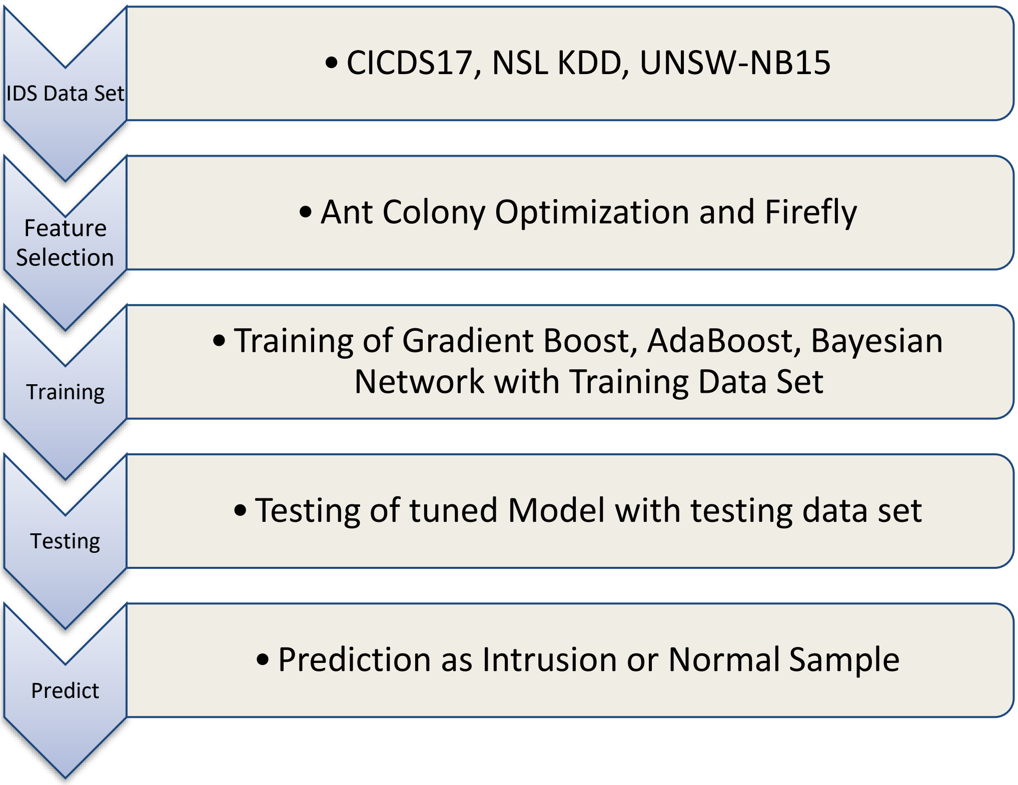

IDS Data Set
• CICDS17, NSL KDD, UNSW-NB15
Feature Selection
• Ant Colony Optimization and Firefly
Training
• Training of Gradient Boost, AdaBoost, Bayesian Network with Training Data Set
Testing
• Testing of tuned Model with testing data set
Predict
• Prediction as Intrusion or Normal Sample

Figure 2. ACO and firefly feature selection and gradient boosting enabled framework for IDS

As shown by example, a BN [23] is a directed acyclic graph (DAG) with a CP distribution at each terminal. This is denoted by the notation $B =< N, A, 0 >$. Each n eN endpoint in the graph represents a domain variable, and each eA edge connecting the n eN endpoints illustrates a probabilistic link between the variables. Therefore, a BN can be used as a classifier because it provides the posterior probability distribution of the classification terminal given the values of other characteristics. The conditional probability, or CP, of one terminal can be estimated using a BN provided the values assigned to the other terminals are known. During the process of learning BN using datasets, researchers make use of terminals to show the features of the datasets.

The idea of a Markov boundary for a terminal will be used in forthcoming research. Markov boundary is defined as a subset of terminals that "isolates" n from the influence of any terminal that lies beyond the boundary. The Markov blanket is one of the Markov borders of the number n. It is formed by the union of the parents of the number n, the children of the number n, and the parents of the children of the number n. When applying a BN classifier to the whole data set, it is quite acceptable to ignore any attributes that fall outside of the Markov blanket. This results in a considerably decreased BN in many situations without compromising the categorization accuracy in any way.

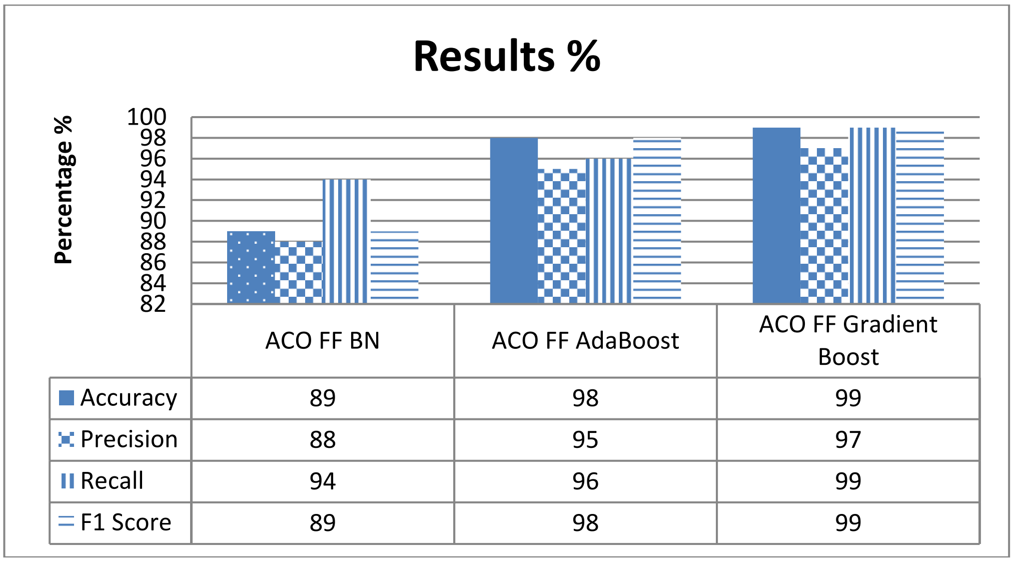
To put it another way, boosting is a means through which pupils who are not doing well academically might improve their performance. Using this strategy means that the dataset that is used to train each subsequent tree is a modified version of the dataset that was used initially. Boosting increases the performance of each individual component of an ensemble by integrating fundamental principles to create it. This is done in order to develop an ensemble. One possible formulation of the ensemble composite hypothesis is: where h is a collection of hypotheses (h1, h2, h3,..., hT).

Gradient boosting is a well-known Boosting technique. The objective of gradient boosting is to generate a function F*(x) that maps x to y in such a way that the expected value of (y, F(x)) is minimized when the joint distribution of all values (y, x) is taken into account. In this context, y refers to the random output or dependent variable, and x=x1, x2, and xn refers to a set of random input variables. One of the numerous qualities that gradient-boosting machines have is their degree of adaptability. A few of the tuning elements that may be altered to fine-tune a system's flexibility are the number of trees, tree depth, learning rate, and sub sampling. Randomization was included into the first version of the gradient boosting machine (GBM) algorithm so that it might perform more effectively. During each iteration of the training process, a subsample of the training data is selected at random from the whole training data set. This approach does not result in the loss of any data. Instead of using the whole training data set, a smaller subsample is used in order to fit the base learner and determine the update that should be made to the model for the current iteration [24]. If the weight of a single instance in the dataset is reliant on the findings of the base classifier for that instance, then the dataset has its own distinct base classification in the AdaBoost decision tree. If they misclassify an instance in subsequent models, the weight of that instance will increase, but if the classification is right, the weight will remain the same [25].

4. RESULT ANALYSIS AND DISCUSSION

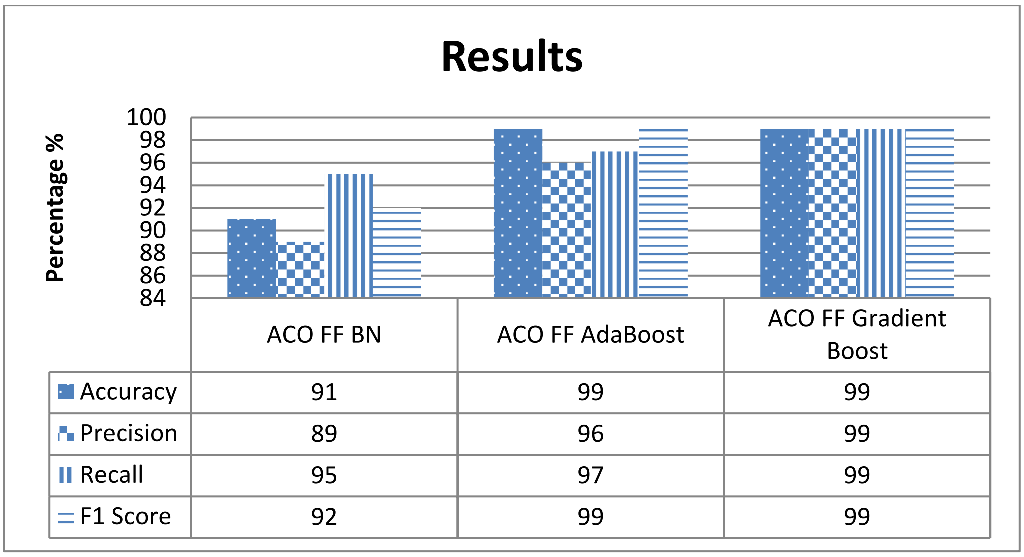
Three datasets NSL-KDD UNSW-NB15, and CICIDS 2017 are used for experimental and evaluation. For NSL-KDD data set, 125,973 samples are used for the training and 22,544 samples are used for testing. NSL KDD data set consists of 42 parameters. For UNSW-NB15, 175,341 samples are used for the training and 82,332 samples are used for testing. UNSW-NB15 data set consists of 49 parameters. For CICIDS 2017, 1,744,184 samples are used for the training and 747,505 samples are used for testing. CICIDS 2017 data set consists of 80 parameters.

Conjunction of ACO and firefly method for selecting features from the intrusion data sets. This optimum feature selection results in increasing classification accuracy. ML techniques like-AdaBoost, gradient boost, and BN are used to classify selected features. Accuracy, precision, recall and F1 parameters are used to evaluate the experimental performance of ML and feature selection techniques. Results for all three data sets are shown in Figures 3-5.



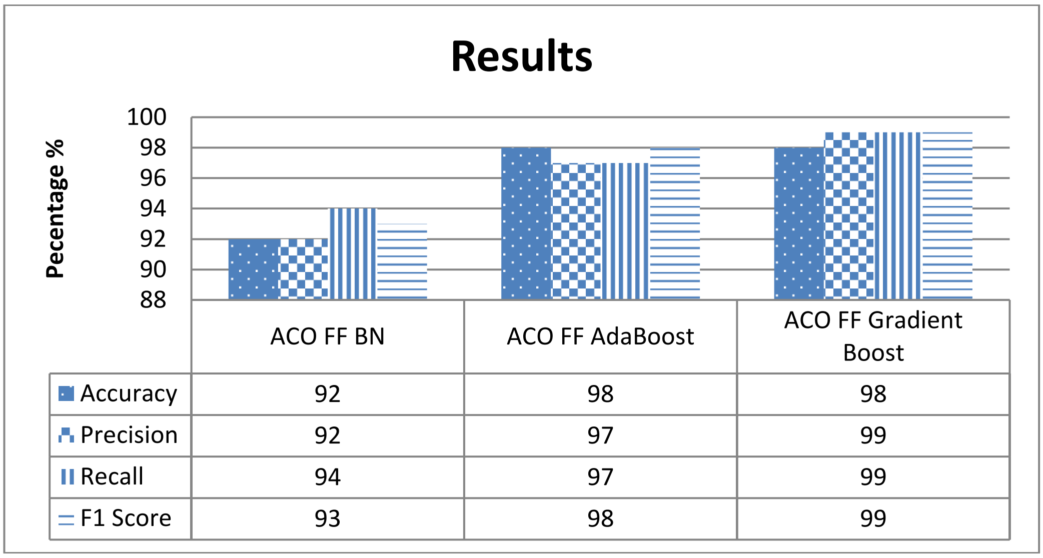
	ACO FF BN	ACO FF AdaBoost	ACO FF Gradient Boost
■ Accuracy	89	98	99
▨ Precision	88	95	97
‖ Recall	94	96	99
= F1 Score	89	98	99

Figure 3. Accuracy, precision, recall, and F1 parameters for NSL-KDD dataset



	ACO FF BN	ACO FF AdaBoost	ACO FF Gradient Boost
Accuracy	91	99	99
Precision	89	96	99
Recall	95	97	99
F1 Score	92	99	99

Figure 4. Accuracy, precision, recall, and F1 parameters for UNSW-NB15 dataset



	ACO FF BN	ACO FF AdaBoost	ACO FF Gradient Boost
Accuracy	92	98	98
Precision	92	97	99
Recall	94	97	99
F1 Score	93	98	99

Figure 5. Accuracy, precision, recall, and F1 parameters for UNSW-NB15 dataset

Where: Accuracy=(true positive+true negative)/(true positive+true negative+false positive+false negative)
Precision=true positive/(true positive+false positive)
Recall=true positive/true positive+false negative)
F1=2*((precision*recall)/(precision+recall))

5. CONCLUSION

An intrusion into a computer network or system is defined as any activity taken with the intention of circumventing such safeguards. The first and most important step in the process of resolving the issue at hand is coming to terms with the fact that there has been a breach in the system's security. Intrusion detection, often known as ID, is the process that is used to locate these unwanted intrusions. The IDS may be broken down into its two component pieces, which are training and testing. During the training phase, a number of different models are developed, each of which is capable of distinguishing between actions that are typical of the dataset and actions that are not typical of the dataset. The models that were developed are scored according to how well they can be classified. This article outlines a method for identifying malicious activity on a computer network. When it comes to picking features from the intrusion data sets, this system makes use of a combination of the ACO approach and the firefly method. The optimal selection of features leads to improved accuracy in categorization. For the purpose of classifying chosen characteristics, ML methods such as AdaBoost, gradient boost, and BN are used. There are three datasets that are utilized for experimental and assessment purposes: NSL-KDD, UNSW-NB15, and CICIDS 2017. Accuracy, precision, recall, and the F1 score are the criteria that are used to evaluate performance. When it comes to intrusion detection and classification, the performance of gradient boost is superior.

ACKNOWLEDGEMENTS

Authors would like to thank Mahakal Institute of Technology, Ujjain, India for the support given during this work.

REFERENCES

[1] M. de Lucia and C. Cotton, “Identifying and detecting applications within TLS traffic,” in *Cyber Sensing 2018*, May 2018, pp. 179–190, doi: 10.1117/12.2305256.

[2] A. Raghuvanshi *et al.*, “Intrusion detection using machine learning for risk mitigation in IoT-enabled smart irrigation in smart farming,” *Journal of Food Quality*, pp. 1–8, Feb. 2022, doi: 10.1155/2022/3955514.

[3] K. Siddique, Z. Akhtar, F. A. Khan, and Y. Kim, “KDD cup 99 data sets: A perspective on the role of data sets in network intrusion detection research,” *Computer*, vol. 52, no. 2, pp. 41–51, Feb. 2019, doi: 10.1109/MC.2018.2888764.

[4] C. Luo, L. Wang, and H. Lu, “Analysis of LSTM-RNN based on attack type of KDD-99 dataset,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, pp. 326–333, doi: 10.1007/978-3-030-00006-6_29.

[5] M. Hafsa and F. Jemili, “Comparative study between big data analysis techniques in intrusion detection,” *Big Data and Cognitive Computing*, vol. 3, no. 1, pp. 1–13, Dec. 2018, doi: 10.3390/bdcc3010001.

[6] J. Kim, C. Sim, and J. Choi, “Generating labeled flow data from MAWILab traces for network intrusion detection,” in *Proceedings of the ACM Workshop on Systems and Network Telemetry and Analytics - SNTA ’19*, 2019, pp. 45–48, doi: 10.1145/3322798.3329251.

[7] P. M. Lutscher, N. B. Weidmann, M. E. Roberts, M. Jonker, A. King, and A. Dainotti, “At home and abroad: The use of denial-of-service attacks during elections in nondemocratic regimes,” *Journal of Conflict Resolution*, vol. 64, no. 2–3, pp. 373–401, Feb. 2020, doi: 10.1177/0022002719861676.

[8] C. Hesselman *et al.*, “The DNS in IoT: Opportunities, risks, and challenges,” *IEEE Internet Computing*, vol. 24, no. 4, pp. 23–32, Jul. 2020, doi: 10.1109/MIC.2020.3005388.

[9] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, “BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset,” *IEEE Access*, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.

[10] F. Salo, A. B. Nassif, and A. Essex, “Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection,” *Computer Networks*, vol. 148, pp. 164–175, Jan. 2019, doi: 10.1016/j.comnet.2018.11.010.

[11] A. Raghuvanshi, U. K. Singh, T. Kassanuk, and K. Phasinam, “Internet of things: Security vulnerabilities and countermeasures,” *ECS Transactions*, vol. 107, no. 1, pp. 15043–15052, Apr. 2022, doi: 10.1149/10701.15043ecst.

[12] A. Rane, A. Waghmare, A. Madhukar, and A. Markad, “Host based internal intrusion detection system,” *Imperial Journal of Interdisciplinary Research*, vol. 2, no. 6, pp. 15–25, 2016.

[13] D. Moon, S. B. Pan, and I. Kim, “Host-based intrusion detection system for secure human-centric computing,” *The Journal of Supercomputing*, vol. 72, no. 7, pp. 2520–2536, Jul. 2016, doi: 10.1007/s11227-015-1506-9.

[14] A. Le, J. Loo, K. Chai, and M. Aiash, “A specification-based IDS for detecting attacks on RPL-based network topology,” *Information*, vol. 7, no. 2, pp. 1–19, May 2016, doi: 10.3390/info7020025.

[15] M. Ahmed, A. N. Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, Jan. 2016, doi: 10.1016/j.jnca.2015.11.016.

[16] M. Blowers and J. Williams, “Machine learning applied to cyber operations,” *Network science and cybersecurity*. pp. 155–175, 2014, doi: 10.1007/978-1-4614-7597-2_10.

[17] A. M. Fawaz and W. H. Sanders, “Learning process behavioral baselines for anomaly detection,” in *2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC)*, Jan. 2017, pp. 145–154, doi: 10.1109/PRDC.2017.28.

[18] Z. Muda, W. Yassin, M. N. Sulaiman, and N. I. Udzir, “K-means clustering and naive bayes classification for intrusion detection,” *Journal of IT in Asia*, vol. 4, no. 1, pp. 13–25, Apr. 2016, doi: 10.33736/jita.45.2014.

[19] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, “A survey of deep learning-based network anomaly detection,” *Cluster Computing*, vol. 22, no. 1, pp. 949–961, Jan. 2019, doi: 10.1007/s10586-017-1117-8.

[20] K. F. Yu, R. E. Harang, and K. N. Wood, “Machine learning for intrusion detection in mobile tactical networks,” in *Cyber Sensing 2017*, May 2017, pp. 31–44, doi: 10.1117/12.2261683.

[21] L. C. Sekhar and R. Vijayakumar, “Feature selection using ant colony optimization and weighted visibility graph,” in *Evolution in Computational Intelligence*, 2021, pp. 17–32, doi: 10.1007/978-981-15-5788-0_3.

[22] E. Ergun and O. Aydemir, “Firefly algorithm based feature selection for EEG signal classification,” in *2020 Medical Technologies Congress (TIPTEKNO)*, Nov. 2020, pp. 1–4, doi: 10.1109/TIPTEKNO50054.2020.9299273.

[23] W. M. D. Dlamini, S. P. Simelane, and N. M. Nhlabatsi, “Bayesian network-based spatial predictive modelling reveals COVID-19 transmission dynamics in Eswatini,” *Spatial Information Research*, vol. 30, no. 1, pp. 183–194, Feb. 2022, doi: 10.1007/s41324-021-00421-6.

[24] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, “A comparative analysis of gradient boosting algorithms,” *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021, doi: 10.1007/s10462-020-09896-5.

[25] S. Rachmadi, S. Mandala, and D. Oktaria, “Detection of DoS attack using AdaBoost algorithm on IoT system,” in *2021 International Conference on Data Science and Its Applications (ICoDSA)*, Oct. 2021, pp. 28–33, doi: 10.1109/ICoDSA53588.2021.9617545.

BIOGRAPHIES OF AUTHORS

Mutyalaiah Paricherla is Assistant Professor in NBKRIST, Vidyanagar Tirupati (A.P), India. His research interest includes machine learning, network security and natural language processing. He is having a teaching experience of 15 years. He can be contacted at email: mutyam2014@gmail.com.

Mahyudin Ritonga is Associate Professor an Arabic Language and Education at the Faculty of Islamic Studies, Muhammadiyah University of West Sumatra. He received the Doctoral degree from the Islamic Studies at Graduate School of Islamic State University SyarifHidayatullah Jakarta. He currently becomes a member of ADRI, IMLA. He works on Arabic Linguistic, Semantics of Arabic Language, Curriculum of Arabic Learning, Strategies of Arabic Language Learning, Research Methodology, and Islamic Education. He can be contacted at email: mahyudinritonga@gmail.com.

Dr. Sandip R. Shinde is Professor and Head of Computer Engineering Department, Vishwakarma Institute of Technology Pune, India. His research interests are deep learning, machine learning, healthcare analytics and network security. He can be contacted at email: sandeep.shinde@vit.edu.

Dr. Smita M. Chaudhari working as an Assistant Professor in the Department of Computer Engineering at Marathwada Mitra Mandal's College of Engineering, Pune, India. She completed her Ph.D. in the field of Computer Science and Engineering from Koneru Lakshmaiah Education Foundation (K. L. University), Vaddeswaram, India. She is in teaching profession for more than 20 years. She has published more than 30 papers in National and International Journals, Conference (Scopus and Web of Science indexing). She has published 3 Books and 2 Patents with 1 granted. Her area of interest includes information security, cloud computing, advanced databases technologies and internet of things. She is a Life member of ISTE, CSI and Member of IEEE. She can be contacted at email: smita.m.c@gmail.com.

Rahmat Linur Lecturer an Arabic Language and Literature at the Faculty of Ushuluddin and Dakwa, college of Islamic state Mandailing Natal of North Sumatra. He received the Master degree from the Islamic Studies at Graduate School of Islamic State Maulana Malik Ibrahim Malan. He can be contacted at email: rahmatlinur@stainmadina.ac.id.

Prof. Abhishek Raghuvanshi is working as head of Department in Department of Computer Science and Engineering in Mahakal Institute of Technology in Ujjain in India. He is having teaching experience of 17 years, research experience of 13 reays and administrative experience of 7 years. His research areas include-machine learning, internet of things security, health care analytics. He is having many publications in SCI and Scopus indexed journals. He has also worked on many governments of India funded research projects. He can be contacted at email: abhishek14482@gmail.com.