International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 01 | January - 2023 Impact Factor: 7.185 ISSN: 2582-3930

Cybersecurity Analysis Using Machine Learning

Shilpa. N
Department of Electronics and telecommunication engineering
Dr. AIT Bangalore
email:Shilpa.n11840@gmail.com

Prof. Yamuna Devi C.R.
Department of Electronics and telecommunication engineering
Dr. AIT Bangalore
email: yamunadevicr.et@drait.edu.in

Asst Prof. Kavitha Narayan K.B
Department of Electronics and telecommunication engineering
Dr. AIT Bangalore
Email: Kavithanarayan.et@drait.ed.in

Abstract

Computers, networks, programs, and data are protected from attacks and unauthorized access, revision, or destruction through a set of technologies and processes known as cybersecurity. Due to the prevalence of sensitive information such as credit card numbers, ATM leg numbers, and so on, cyber security is now a major concern in the software industry. got addressed and pasted into publicly accessible websites. As a result, we selected this concept to investigate novel approaches to completely eliminating cyber security issues. Our system, which we are proposing, will be based on machine literacy and take the input in the form of a dataset like KDD Cup 99 or another dataset that is helpful in identifying cyber security issues. We will examine the fine points of three distinct bracket algorithms, including the SVM algorithm, Random Forest, and the Random Plus algorithm, which is based on the Random Forest algorithm. The achieved delicateness will demonstrate that our proposed algorithm will perform significantly better than the other two algorithms. The performance of the dataset processing will be covered, and the entire setup will be mounted in the pall garçon, which is a component of structure as a Service (IaaS).

Keywords: Cloud Computing, Cyber Security, Machine Learning, Cyber Security Dataset, IaaS Cloud, Classification Algorithm.

I) INTRODUCTION

A computer security system and a network security system make up a network security system. Firewalls, antivirus software, and intrusion detection systems (IDS) are all part of these systems. The service provided by IDSs identifies, controls, and discovers unauthorized system geste such as operation, copying, revision, and destruction. The Internet is altering how people study and work, but it also exposes us to fewer and fewer security risks as it becomes increasingly integrated into everyday life. A critical issue that needs to be resolved right away is how to identify colorful network attacks, particularly attacks that have never been seen before.

Both external and internal intrusions constitute security breaches. For IDSs that are misuse-ground, there are three primary types of network analysis: anomaly-ground, hand-ground, and mongrel. Abuse-constructed discovery methods aim to distinguish known attacks based on their signatures. They don't cause a lot of false warnings because they are secondary to known attacks.

Anomaly-based methods for understanding normal network and system behavior and recognizing anomalies as deviations from normal behavior Their capacity to distinguish zero-day attacks makes them appealing. Another advantage is that standard exertion biographies are tailored to each system, operation, or network, making it difficult for bushwhackers to determine which conditions they can

negotiate unnoticed. In addition, the data that can be used to define the signatures for abuse sensors are the anomaly-based alerts (new attacks). A major drawback of anomaly-based approaches is the possibility of high false alarm rates due to the fact that initially unseen system actions can be categorized as anomalies.

The Knowledge Discovery and Data Mining (KDD Cup 99) dataset has become one of the most widely used datasets for evaluating the effectiveness of intrusion detection systems over the past ten years. Researchers use a variety of machine literacy techniques to learn how to recognize similar attacks because there are numerous types of attacks. Still, in the previous work on using machine literacy in IDS experiments, the researchers used certain bracket algorithms to find different kinds of colorful attacks. This work has reached maturity thanks to experimenters with strong computer network security skills but limited knowledge of machine literacy algorithms. As a result, the attempts have only used the machine learning algorithm as a "tool" to find attacks. However, many important aspects of its use and the necessary fine-tuning required for using similar tools have not been given the importance they deserve. The number of training samples used to train minor attacks—attacks that are rare and do not have enough samples to train the classifier model—is significantly lower than the number of samples used to train major classes, which is why standard methods for using machine literacy classifiers to classify data in imbalanced datasets frequently fail. Examples include attacks of the minor classes R2L (Remote to Original) and U2R (Stoner to Root), which frequently exhibit poor discovery rigor. This problem is the focus of the investigation in this thesis.

A review of machine literacy (ML) styles for cyber-security operations is the focus of this paper. In network intrusion discovery, each system's ML styles and some operations are described. It focuses on network security ML technologies, ML styles, and their descriptions. The keywords "machine literacy" and "cyber" are the focus of our research, which will be examined on Google Scholar. Particularly because they describe the current practices, the most recent hot papers are handed down to me.

The study in this paper aims to determine whether the proposed system is based on machine literacy and takes data from datasets like KDD Cup 99 or other datasets useful for identifying cyber security issues. We will examine the finer points of three distinct bracket algorithms—the SVM algorithm, the Random Forest algorithm, and the Random Plus algorithm—all of which are based on the Random Forest algorithm. The achieved delicateness will demonstrate that our proposed machine literacy algorithm will perform significantly better than the other two.

RELATED WORK

This paper does not cover every approach to network anomaly discovery; rather, it only focuses on ML methods. Still, hand-on and cold-blooded approaches are shown in addition to anomaly discovery.

SalihaYeşilyurt in 2021; HidayetTakç;(1) Problems with language and communication may indicate autism, a widespread experimental disability. Even though webbing tests typically take a lot of time and money, they are always used to diagnose this kind of illness. In recent times, machine literacy methods have been utilized frequently for this purpose due to their effectiveness. An empirical evaluation of the overall performance of eight distinct machine literacy (ML) algorithms detecting autism condition is carried out using four distinct standard datasets from QCHAT, AQ- 10 adult webbing tests, and AQ- 10 child. Using criteria for perceptivity, perfection, particularity, and bracket delicacy, we estimate their performances. The experimental marks demonstrate that stylish marks are produced by a SVM-based representation called C- SVC. In addition, all datasets result in a 100 percent reservation rate for C-SVC donation measures. Multiple variable regression models were presented alternately. On the other hand, the worst issues are found using a decision tree foundation approach and a C4.5 algorithm.

According to Claudio Gallicchio, Alessio Micheli, D. José Martn Guerrero, and Emilio Soria Olivas (2017, p. 2), machine literacy (ML) models have always included randomness in some capacity. The use of randomness has changed over time, no longer as a specific and garnishment improvement in extremely meticulous aspects of a model but rather as the primary theoretical basis that supports some Machine-literacy (ML) styles, such as, familiar arbitrary timbers. Since its inception, randomness has evolved into a large number of models in the Neural Network (NN) neighborhood, which have recently been used primarily for efficiency projects. However, bias brought about by the use of NNs with arbitrary weights requires additional investigation, particularly in light of recent developments in the fields of deep NNs, dynamical systems (intermittent NN), and NNs for learning in well-planned fields.

The movie "YongSik Kim in 2020; Thus, Min; YouKyung Kim"(3) Iterative primer testing on scripts and cases has been used in conventional IT systems to release newly developed software and systems. The time and financial constraints of IT systems prevent these conventional test cases from incorporating implicit scripts and actual cases. As a result, it's possible that we won't be able to completely eliminate all implicit errors before go-live, which could result in unanticipated failures that could seriously harm guests and IT design service provider. To demonstrate a genuine automated software testing strategy known as "Perfec- Twin," this paper provides colorful real-world examples. Perfec- Twin outperforms forenamed flaws of conventional testing and may nearly eliminate all implicit errors before going live by running new and old systems side by side and automatically validating the new system against the facts of the old system in real time.

"Rudolf Ramler and Claus Klammer" (2017, p. 4): "Test robotization is crucial in situations of rapid-fire nimble development." The primary goal is to reduce the amount of manual testing labor and cut down on test prosecution cycles. We went beyond test robotization and used a GUI-based operation designed for significant assistance to generate test cases. In the paper, the transition from manual investigative testing to automated GUI test-case generation is explained. The following are the three primary lessons to be learned: automated test case generation does not eliminate testing issues; instead, it shifts its focus to writing test-case appendages and checks. The difficulty of evaluating the generated test results limits the practical operation of automated test generation. The requirement for sky-scraping position GUI testing is often overlooked by the test-case robotization aggregate that is desired for agile development. The study outlines how test-case generation can be used effectively in real-world business assignments, but it also identifies a number of unsolved issues that test robotization research needs to address in the future.

The following individuals were mentioned: "Sebastian Mayer, Laura von Rueden, Katharina Beckh, BogdanGeorgiev, Annika Pick, Rajkumar Ramamurthy, JochenGarcke, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Michal Walczak, Julius Pfrommer, Christian Bauckhage, and Jannis Schuecker" (2017, p. 5) Despite its considerable success, machine literacy can have significant limitations when faced with a lack of training data. When redundant information is incorporated into the training process, the concept of knowledgeable machine literacy (ML) emerges. In this paper, we provide a structured overview of numerous approaches to this topic. We provide a paradigm that separates informed machine literacy from

traditional machine literacy (ML), outline its abecedarian characteristics, and define it. By well-informed machine literacy (ML) methods, a title that serves as a shell for bracket is present in attendance. It incorporates the birth, illustration, and objectification of knowledge into the machine literacy (ML) process. In support of this title, we look at related pieces of literature and discuss how algebraic equations, logical rules, and simulation issues are used in learning systems to represent colorful knowledge. Using our taxonomy as a foundation, this analysis of multiple papers reveals important aspects of informed machine literacy.

Table 1: Summary of related works.

Reference	Objective	Algorithm
1	This work uses four separate benchmark datasets that comes from QCHAT, AQ-10 adult screening tests & AQ-10 child and conducts an empirical evaluation of the performances on the whole eight distinguished machine learning (ML) algorithms detecting autism condition.	C4.5 algorithm
2	Randomness has forever been in attendance in one/previous form in Machine-Learning (ML) models. The final few years have seen a change of role in the make use of randomness, which is no longer a specific & ornament enhancement in extremely scrupulous aspects of a model.	Random Forest algorithm
3	This paper provides various real-world examples to illustrate a genuine transaction based automated software-testing approach called "Perfec-Twin".	Manual Testing
4	The test-case automation pyramid wished-for for agile-development tends to underestimate the necessitate for sky-scraping level GUI testing.	GUI Testing
5	We review interrelated literature & discuss the application of various knowledge representations in learning systems, including algebraic equations, logical rules & simulation outcomes.	SVM algorithm

PROPOSED SYSTEM

ML is a subfield of AI that has a lot in common with computational statistics, which also focuses on using computers to harvest timber. It has strong ties to fine optimization, which contributes to the field by distributing styles, propositions, and operation disciplines. Although unsupervised literacy is the ultimate subfield that focuses additional on experimental data analysis, ML and data mining are rarely combined. Unsupervised machine learning (ML) can also be used to discover meaningful anomalies and learn and establish birth behavioral biographies for colorful realities. Arthur-Samuel, the founder of ML, referred to ML as "a field of literacy that gives computers the capacity to learn without being clearly programmed." ML primarily focuses on bracket and retrogression based on pre-learned structures based on the training data.

Long-running datasets, a lack of information, and erratic grouping figures all plague existing datasets. Even though the data can be improved after processing, there is insufficient data volume and poor data management. As a result, the development of network intrusion discovery datasets containing substantial amounts of data, a variety of content, and balanced sample figures of attack orders rises to the top of the intrusion discovery priority list.

Our system, which we are proposing, will be based on machine literacy and take the input in the form of a dataset like KDD Cup 99 or another dataset that is helpful in identifying cyber security issues. We will examine the finer points of three distinct bracket algorithms—the SVM algorithm, the Random Forest algorithm, and the Random Plus algorithm—all of which are based on the Random Forest algorithm. The achieved delicateness will demonstrate that our proposed machine literacy algorithm will perform significantly better than the other two.

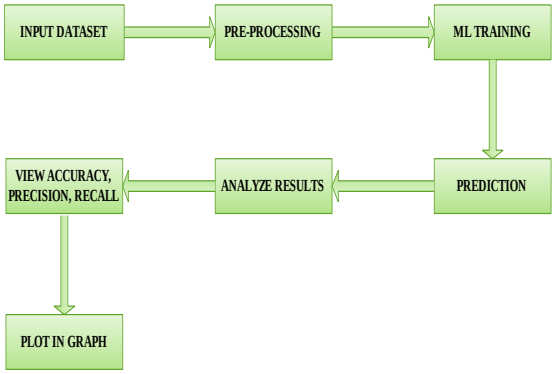

Fig 1: Architecture Diagram

A) SVM Algorithm

SVM is an instigative algorithm with relatively unpretentious generalizations. The classifier uses a hyperplane with the major quantum of the periphery to separate the data points. Because of this, an SVM classifier is recognized as a judicial classifier as well. SVM identifies the best hyperplane that aids in the classification of novel data points.

SVM has the same high precision as new classifiers like logistic retrogression and decision trees. It is praised for the ability of its kernel to grasp nonlinear input spaces. Face discovery, intrusion discovery, the bracket of emails, news papers, and web runners, the bracket of genes, and handwriting recognition are just a few of the operations that involve it.

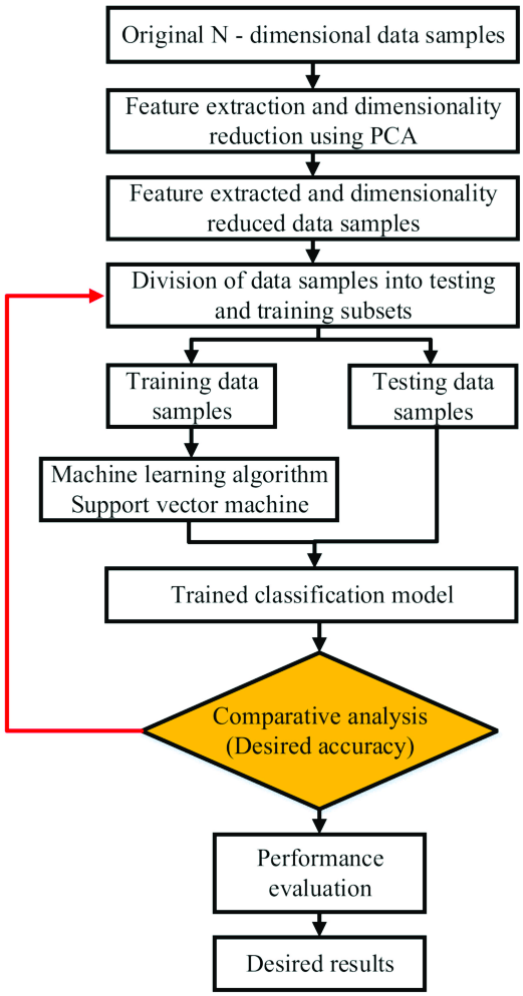

Figure 2: SVM Algorithm Flow Chart

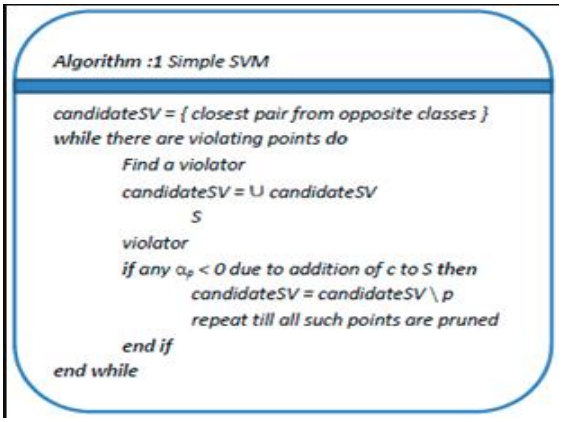

Figure 3: SVM Pseudocode

B) Random Forest Algorithm

A supervised literacy algorithm is Random Timbers. Both the bracket and the retrogression can be handed down simultaneously. Similar to the most adaptable and easy-to-use operation algorithm, Trees constitute a timber. It is believed that the more recent the trees, the more robust and redundant the wood will be.

On random nominated data samples, random trees generate decision trees, receive validation from each tree, and vote for the stylish result. It likewise bears the cost of an enticing decent record of the point importance. Random timbers has a variety of operations that are comparable to recommendation machines, image brackets, and point selection. It is possible to disqualify pious loan applicants, denounce fraudulent behavior, and forecast conditions. The Boruta algorithm, which selects significant features from a dataset, is based on its deceitfulness.

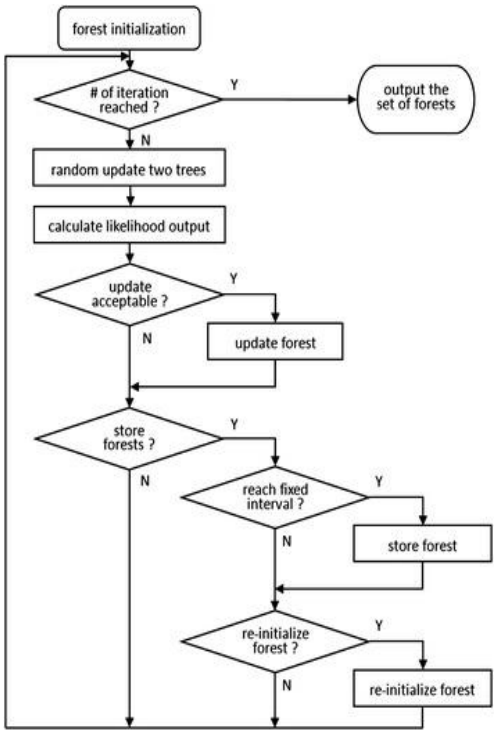

Fig 4: Flowchart of Random Forest Algorithm

Algorithm 1: Pseudo code for the random forest algorithm

To generate c classifiers:
for $i = 1$ to c **do**
  Randomly sample the training data D with replacement to produce D_i
  Create a root node, N_i containing D_i
  Call BuildTree(N_i)
end for

BuildTree(N):
if N contains instances of only one class **then**
  return
else
  Randomly select x% of the possible splitting features in N
  Select the feature F with the highest information gain to split on
  Create f child nodes of N, N_1,..., N_f, where F has f possible values (F_1, ... , F_f)
  for $i = 1$ to f **do**
    Set the contents of N_i to D_i, where D_i is all instances in N that match
    F_i
    Call BuildTree(N_i)
  end for
end if

Fig 5: Random Forest Algorithm Pseudocode

Additionally, Random-Trees provides a respectable point selection index. With the model, Scikit Learn adds a redundant variable that shows the relative character or donation of each point in the validation. During the training phase, it automatically calculates each point's significance score. Additionally, it scales the bearing down until the sum of all scores equals 1. You can use this score to select the features that are most essential and eliminate the bones that are not.

The "Gini-Importance" or "mean drop in contamination" (MDI) method is used by Random-Tree to decipher the character of each point. Gini, which means "the whole drop in knot contamination," is also revered. This is the extent to which the model's fit or delicacy decreases when a variable is removed. The new significance of the variable increases with the size of the drop. The mean drop is then an important variable selection parameter. The variables' complete explicatory authority can be gasped by the Gini indicator.

C) Random Plus Algorithm

The Supervised Machine Learning Algorithm Random Plus is frequently utilized in bracket and regression problems. It takes maturity votes for bracket and normal in the event of retrogression and constructs decision trees from various samples. The Random Integration Algorithm's ability to handle a data set with both categorical and nonstop variables, as in the case of bracket, is one of its most important characteristics. For bracket problems, it produces superior results.

A measure is used to manage the diversity of existing and new neighbor results during the generation of neighbor results. During the construction of new neighbors, the change from the current point is multiplied by The measurement is a sine function.

$$\alpha = 121 + \text{sini}\theta\pi\text{Nneigh} \qquad \text{Eq 1}$$

Here i am the index of neighbor, Nneigh is entire number of neighbor solutions which is being produced at each iteration, & θ is a parameter that controls the oscillation period of α.

An objective principle that intended based on sigmoid function given by

$$Sk=11+e-\sigma k-kcenter\times M \qquad \text{Eq 2}$$

$$fkx-fk-\Gamma xfk-\Gamma x<\delta \qquad \text{Eq 3}$$

Where δ is the ratio of alteration in objective-function value, Γ = ηM, and η is the fraction of the maximum iterations (M) by which the modify in the objective-function is evaluated. As per this stopping measure, if the enhancement over Γ generations is no longer than a threshold (δ), continuation of further iterations can be in-effective& search should be terminated.

Input: X=Training data, R=Total Features, r=members of features, T=Amount of tress

Output: Input data bagged class tag

Start

1. While a distinct tree in Forest T:

i. Perform Tabu search algorithm to choose a best sample M with size N out of training data.

ii. Build the tree Bt through iteratively repeating the dissimilar steps regarding tree individual node.

a. Choose randomly r from R

b. Select optimal between f.

c. Split node.

2. When the creation of T trees is carried out, examples of test data will be sent to individual tree & allocation of class tag on basis of majority votes will be carried out

End

Fig 6: Random Plus Pseudocode

II) PERFORMANCE VALIDATION

A) The delicacy and response time of Support Vector Machine (SVM), Random Forest, and Random Plus algorithms are compared in the result analysis section. The Random Integration Algorithm is broken down into Support Vector Machine (SVM) and Random Forest Algorithm in the tables and graphs that follow. The Random Forest Algorithm, which we used as the foundation for the Random Plus Algorithm, will always be inferior to the recently proposed algorithm.

B) Implementation setup

With the help of provided dataset for training, 30% of dataset is separated out for testing and remaining 70% is meant for training. For all three below mentioned algorithms their process is the same one.

Table 1 shows the accuracy using Random Plus algorithm is 92 percentages at the 500th iteration whereas the accuracy of Random Forest and Support Vector Machine (SVM) algorithm is 76% and 75%.

Table 1: Summary of different type of algorithm accuracy.

Algorithm	**500 Iterations**	**500+ Iterations**
SVM	75	78
Random Forest	76	82
Random Plus	92	97

Table 1 shows the accuracy using Random Plus algorithm is 92 percentages at the 500th iteration whereas the accuracy of Random Forest and Support Vector Machine (SVM) algorithm is 76% and 75%.

Table 2: Summary of different type of algorithm Precision.

Algorithm		**500 Iterations**	**500+ Iterations**
SVM		70	72
Random Forest		73	85
Random Plus		91	96

C) The results of the production gap

Fig. 7 provides detailed Comparison of accuracy spans between different prediction methods and Random Plus algorithms. The diagram shows that Random Plus provides better results than comparable prediction methods. However, the proposed Random Plus algorithm is efficient, with a minimum the running time and producing more accuracy.

Algorithm	**500 Iterations**	**500+ Iterations**
SVM	75	78
Random Forest	76	82
Random Plus	92	97

Table 3: Accuracy Calculation.

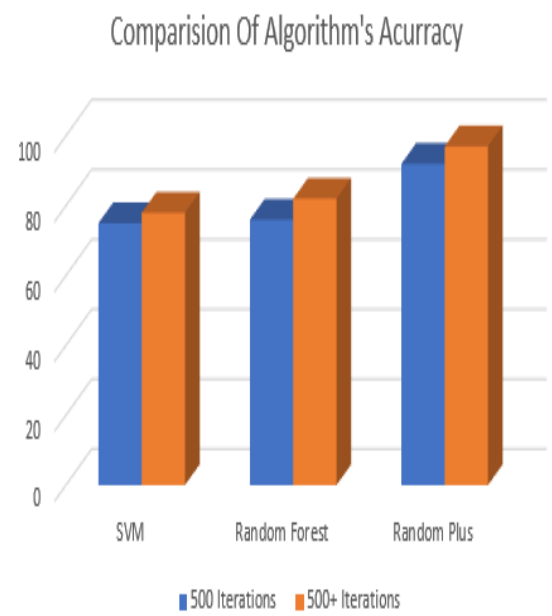

Fig 7: Comparative results analysis of Accuracy.

Analysis of results in Precision

Figure. 8 Provides a detailed comparison of the Precision of different programs methods and Random Plus algorithms. This figure shows that Random Plus gives better results than comparable planning algorithms. Keep in mind that the SVM and Random Forest algorithms require a maximum Precision of 75% and 76%, respectively, when measuring the results of very large dataset. On the other hand, the Random Forest algorithm requires a competitive precision of 73%. However, the proposed Random Plus algorithm is providing maximum of 91% precision.

Table 4: Precision Calculation.

Algorithm	**500 Iterations**	**500+ Iterations**
SVM	70	72
Random Forest	73	85
Random Plus	91	96

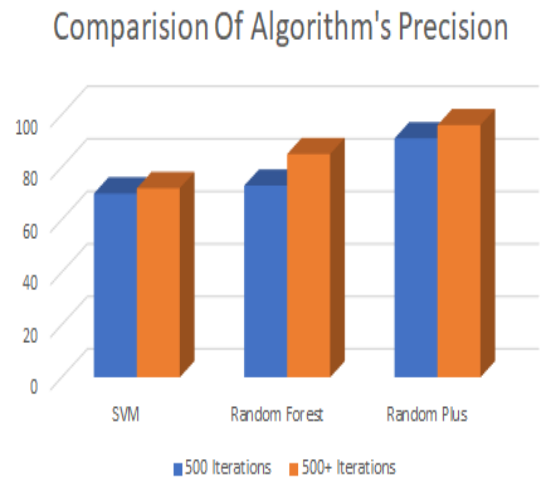

Fig 8: Comparative results analysis in terms of Precision.

VI) Conclusion

Trials are used to determine the effectiveness of the proposed frame by applying it to a case study. These studies' findings are encouraging because they demonstrate that an automated attack discovery process can still achieve a 100 percent accuracy rate in a shorter amount of time. The arbitrary integration method works well when the length of the tree traversal takes a long time. because the length of the cut causes an automatic increase in the degree of the traversal standard. The settings of the Random integration algorithm will be altered in future improvements to the suggested work to make them suitable for both private and cold-blooded hospice environments.

The arbitrary integration approach predicts the affair cost more accurately and delicately when handling the bracket process on pall waiters. Additionally, the success and failure rates of the categorization procedure can be examined by expanding the Random Integration Algorithm. Methods for unborn categorization ought to be investigated in order to guarantee high levels of efficacy and delicateness when implemented in pall systems. In the future, characteristics of the arbitrary integration style, in addition to bracket position, can be investigated to improve delicateness and effectiveness in pall environments.

VII) REFERENCES

[1] J. N. Goetz, A. Brenning, H. Petschko, and P. Leopold, “Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling,” Comput. Geosci., vol. 81, no. 3, pp. 1–11, 2015.

[2] R. P. Lippmann, D. J. Fried, I. Graf, and J. W. Haines, “Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation,” in DARPA Information Survivability Conference and Exposition, 2000. DISCEX ’00. Proceedings, 2000, pp. 12–26 vol.2.

[3] M. Soni, M. Ahirwa, and S. Agrawal, “A Survey on Intrusion Detection Techniques in MANET,” in

International Conference on Computational Intelligence and Communication Networks, 2016, pp. 1027–1032.
[4] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating Computer Intrusion Detection Systems:A Survey of Common Practices," Acm Comput. Surv., vol. 48, no. 1, pp. 1–41, 2015.
[5] E. Viegas, A. O. Santin, A. França, R. Jasinski, V. A. Pedroni, and L. S. Oliveira, "Towards an Energy-Efficient Anomaly-Based Intrusion Detection Engine for Embedded Systems," IEEE Trans. Comput., vol. 66, no. 1, pp. 163–177, 2017.
[6] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "Review: A survey of intrusion detection techniques in Cloud," J. Netw. Comput. Appl., vol. 36, no. 1, pp. 42–57, 2013.
[7] S. Revathi and A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection," in International Journal of Engineering Research and Technology, 2013.
[8] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Comput. Netw., vol. 51, no. 12, pp. 3448–3470, 2007.
[9] C. N. Modi and K. Acha, "Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review," J. Supercomput., vol. 73, no. 3, pp. 1–43, 2016.
[10] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Commun. Surv. Tutor., vol. 18, no. 2, pp. 1153–1176, 2016.
[11] G. Kim, H. Yi, J. Lee, Y. Paek, and S. Yoon, "LSTM-Based System-Call Language Modeling and Robust Ensemble Method for Designing Host-Based Intrusion Detection Systems," arXiv:1611.01726, 2016.
[12] M. Z. Alom, V. R. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in Aerospace and Electronics Conference, 2016, pp. 339–344.
[13] K. Alrawashdeh and C. Purdy, "Toward an Online Anomaly Intrusion Detection System Based on Deep Learning," in IEEE International Conference on Machine Learning and Applications, 2017, pp. 195–200.
[14] N. Gao, L. Gao, Q. Gao, and H. Wang, "An Intrusion Detection Model Based on Deep Belief Networks," in 2014 Second International Conference on Advanced Cloud and Big Data, 2014, pp. 247–252.
[15] G. Zhao, C. Zhang, and L. Zheng, "Intrusion Detection Using Deep Belief Network and Probabilistic Neural Network," in IEEE International Conference on Computational Science and Engineering, 2017, vol. 1, pp. 639–642.
[16] K. Rai, M. Syamala, Devi Professor, and A. Guleria, "Decision Tree Based Algorithm for Intrusion Detection," vol. 07, no. 4, pp. 2828–2834, 2016.
[17] M. Modinat, A. Abimbola, B. Abdullateef, and A. Opeyemi, "Gain Ratio and Decision Tree Classifier for Intrusion Detection," Int. J. Comput. Appl., vol. 126, no. 11, pp. 975–8887, 2015.
[18] C. Azad and V. K. Jha, "Genetic Algorithm to Solve the Problem of Small Disjunct In the Decision Tree Based Intrusion Detection System," vol. 7, no. 8, pp. 56–71, 2015.
[19] S. Puthran and K. Shah, "Intrusion Detection Using Improved Decision Tree Algorithm with Binary and Quad Split," in International Symposium on Security in Computing and Communication, 2016, pp. 427–438.
[20] A. O. R. G. Jimoh, "Anomaly Intrusion Detection Using an Hybrid Of Decision Tree And K-Nearest Neighbor," vol. 2, no. 1, pp. 67–74, 2015.
[21] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," in International Conference on Platform Technology and Service, 2016, pp. 1–5.
[22] R. B. Krishnan and N. R. Raajan, "An intellectual intrusion detection system model for attacks classification using RNN," Int. J. Pharm. Technol., vol. 8, no. 4, pp. 23157–23164, 2016.
[23] Q. Tan, W. Huang, and Q. Li, "An intrusion detection method based on DBN in ad hoc networks," in International Conference on Wireless Communication and Sensor Network, 2016, pp. 477–485.
[24] M. Nadeem, O. Marshall, S. Singh, X. Fang, and X. Yuan, "Semi-Supervised Deep Neural Network for Network Intrusion Detection," KSU Proc. Cybersecurity Educ. Res. Pract., Oct. 2016.
[25] C. L. Yin, Y. F. Zhu, J. L. Fei, and X. Z. He, "A Deep Learning Approach for Intrusion Detection using Recurrent Neural Networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.
[26] D. Moon, H. Im, I. Kim, and J. H. Park, "DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks," J. Supercomput., vol. 73, no. 7, pp. 2881–2895, 2017.
[27] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," Clust. Comput., pp. 1–13, 2017.
[28] S. Jo, H. Sung, and B. Ahn, "A Comparative Study on the Performance of Intrusion Detection using Decision Tree and Artificial Neural Network Models," vol. 11, no. 4, pp. 33–45, 2015.
[29] Y. Ding, S. Chen, and J. Xu, "Application of Deep Belief Networks for opcode based malware detection," in International Joint Conference on Neural Networks, 2016, pp. 3901–3908.
[30] R. C. Staudemeyer, "Applying long short-term memory recurrent neural networks to intrusion detection," vol. 56, no. 1, pp. 136–154, 2015.