ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 11, Issue, 04, pp.3283-3290, April, 2019

DOI: https://doi.org/10.24941/ijcr.35210.04.2019

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

RESEARCH ARTICLE

SR-MLC: SCALABLE RESILIENCE MACHINE LEARNING CLASSIFIERS APPROACH IN CYBER SECURITY

Anil Lamba and *Natasha Dutta

Department of Computer Science, Charisma University, Turks and Caicos Islands

ARTICLE INFO

Article History:
Received 11th January, 2019
Received in revised form
17th February, 2019
Accepted 14th March, 2019
Published online 30th April, 2019

Key Words:
Resilience, Cluster, Cloud, Cyber Security, Artificial Intelligence, Machine Learning, Network, Analytics, Classifiers, Cyber Attackers.

Corresponding author:* **Natasha Dutta

ABSTRACT

Cyber resilience is a rapidly emerging viewpoint that is attaining recognition. Unfavourable Cyber-attacks are those that oppositely influence the availability, integrity or confidentiality of IT network systems and related services and information. Prior research works have carried on data manipulation by an opponent as a concern, but their works failed to generalize the test cases. Many concentrated on devising attack vectors opposite to specific machine learning algorithms and applications, such as the Support Vector Machine (SVM) classifier. In our proposed work, an independent approach on resilience evaluation and the construction of adversary resilient classifiers using Cluster Tree Map (CTM) Algorithm is done. All data types in the domain of Cyber Network data analytics are focussed. The objective is to make an awareness of any such method capable of correctly modelling the creativeness and skill of cyber attackers and thereby developing unsupervised learning model. Better expected accuracy is attained by using Scalable Resilience Machine Learning Classifiers (SR-MLC).

Citation: Anil Lamba and Natasha Dutta, **2019.** "SR-MLC: Scalable Resilience Machine Learning Classifiers Approach in Cyber Security", *International Journal of Current Research*, 11, (04), 3283-3290.

INTRODUCTION

Machine learning, according to its, is a field of software engineering that developed from examining design acknowledgment and computational learning hypothesis in artificial intelligence. It is the learning and working of algorithms that can gain from and make expectations on informational indexes. These systems work by development of a model from precedent contributions to request to settle on information driven forecasts or decisions as opposed to adhering to firm static program instructions (Simon, 2015) Machine learning methods have been connected in numerous zones of science because of their extraordinary properties like flexibility, versatility, and potential to quickly adapt to new and obscure difficulties. Cyber security is a quickly developing field requesting a lot of consideration on account of surprising advances in social networks, web technologies, cloud and mobile environment, online banking, smart grid, and so forth. Various machine learning strategies have been effectively made to address such far reaching issues in cyber security. With the quick evolvement of web and versatile advancements, attack techniques are additionally ending up increasingly refined in several frameworks and avoiding actual signature-based methodologies. Machine learning procedures offer potential solutions that can be utilized for settling such difficult and complex circumstances because of their capacity to adjust rapidly to new and obscure conditions. Different machine learning techniques have been effectively addressed boundless issues in PC and data security (Ford, 2014).

Machine-learning algorithms are utilized to explain a regularly expanding scope of grouping issues in recognition, vision, investigation and surmising over the whole range of computing stages (Dubey, 2005). Machine learning algorithms work in two stages: training and testing. In training, decision models are developed dependent on a marked training informational index. In testing, the model is connected to categorize new input cases (Venkataramani *et al.,* 2015). The utilization of machine learning algorithms for cyber security purposes offers ascend to inquiries of antagonistic strength, to be specific: Can we evaluate the exertion expected of an unfavourable to control a framework that depends on machine learning systems? Could the antagonistic flexibility of such frameworks be formally demonstrated and assessed? Would we be able to measure this strength with the end goal that distinctive frameworks can be looked at utilizing empiric measurements? Past works have shown how an unfavourable one can control a framework dependent on machine learning strategies by changing a portion of its information sources. Although, similarly little work has stressed the production of a formal strategy for estimating and the comparing adversarial resilience of various machine learning models to these changes. Clustering is an exceptionally valuable information exploratory machine learning instrument that enables us to comprehend heterogeneous information by gathering information with comparative attributes dependent on a few criteria. Famous graph partitioning methods, for example, the Girvan– Newman algorithm (Girvan *et al.,* 2002), sparsest-cuts (Shi, 2000), spectral partitioning (Alpert, 1999), and general

conductance based techniques (identified with spectral techniques by means of Cheeger's inequality) (Chung, 1997) might be seen as taking care of an edge-based resilience issue with respect to a graph while at the same time yielding the components coming about because of the expulsion of the critical edge set as the arrangement of clusters. In our proposed work, Cluster Tree Map (CTM) algorithm utilizing structured data from resilience data is created to concentrate on all data types in the zone of Cyber Network information investigation which in turn is utilized to group and classify the resilience data in the successful way. The reason for the proposed work is to make an unsupervised learning model by making a free methodology on the involvement of content experts to estimate feature manipulation costs. An independent availability on resilience evaluation and the construction of adversary resilient classifiers is performed utilizing CTM Algorithm. Thus an awareness of any such method is made capable of correctly modelling the creativeness and skill of cyber attackers. The objectives and our contributions in the proposed work are

- To make an independent method on the involvement of content experts to evaluate feature manipulation costs.
- To make an independent presence on resilience estimation and the construction of unfavourable resilient classifiers.
- To make numeric techniques depend on the opinions of experts might seem preferable.
- To make an awareness of any such method capable of correctly modelling the innovativeness and cyber attackers' skill.
- To utilize unsupervised learning model and all these are done using CTM algorithm.

Rest of the paper is organized as follows. Section 2 presents related work. Cluster Tree Map algorithm is briefly explained in section 3. In section 4, attacks exploiting machine learning systems are described briefly. Results are discussed in section 5. The paper is concluded in section 6 with future directions.

Literature Review

Anna L. Buczak described data mining (DM) and machine learning (ML) techinques for cyber analytics related to attack detection. Some popular cyber data sets utilized in ML/DM Special emphasis was made on the utilization of various ML and DM techniques in the cyber domain, both for detection of misuse and anomaly. As a result, it was impossible to make one recommendation for each method, based on the type of attack the system is supposed to detect (Buczak and Guven, 2016). Nguyen *et al.* (2008) explained ML techniques for categorization of traffic in Network. The methods explained did not depend on already well-known port numbers but on statistical traffic features. The processes are divided and reviewed as per their options of ML strategies. The promising outcomes of ML-based IP traffic categorization opened several new avenues for related research domains, like the application of ML in intrusion detections, anomaly detection in user data and control, routing traffic, and building network profiles for proactive network real-time monitoring and management. Teodoro *et al.* (2009) focussed on anomaly intrusion techniques in network. Statistical, knowledge-based, and machine-learning approaches were presented, but their study does not present a full set of machine-learning techniques. Wu *et al.* (2010) concentrated on Computational Intelligence methodologies and their applications to attacker detection. Techniques namely Swarm Intelligence Artificial Neural Networks (ANNs), Evolutionary Computation, Fuzzy Systems, and Artificial Immune Systems are explained in detail. Since only Computational Intelligence methods are explained, several ML/DM methods like decision trees, clustering, and rule mining have not been incorporated. Its characteristics, like adaptation, high operational speed, fault tolerance, and error resiliency in the area of noisy information, fit the requirement of building a good system of detecting intrusion. Revathi and Malathi (Revathi, 2013) concentrated on machine-learning intrusion techniques. The authors presented an explanative set of machine-learning algorithms on the NSL-KDD intrusion detection dataset, but their study only incorporated a misuse detection context. In contrast, this work explained not only misuse detection but also anomaly detection. The outcomes depicted that NSL-KDD dataset is much ideal for comparison of various kind of intrusion detection models. Buczak and Guven (2016) focussed on machine-learning techniques and their utilization in intrusion detection. Algorithms like Neural Networks, Genetic Algorithms, Support Vector Machine, Bayesian Networks, Fuzzy Logics, and Decision Tree were described in detail.

Accuracy, time for categorizing an unknown instance with a trained model, and complexity, understandability of the final solution (classification) of each ML or DM method provided better outcomes. Sahoo *et al.* (2017) introduced the formal procedure of Malicious Detection of URL as a machine-learning process and divided and previewed their contributions that address different dimensions of the problems like feature representation and algorithm design. However, they did not explain the technical details of the algorithm. Pervez and Farid (16) proposed a filtering algorithm based on SVM classifier to select multiple intrusion classification performances on the NSL-KDD intrusion detection dataset. The method maintained the classification accuracy of the SVM classifier but it uses a reduced set of input features from training data.

METHODOLOGY

Dynamic analysis report database is taken at first. Dynamic analysis of malware incorporates the executing process of malware, monitors its characteristics, and generates a profile. It detects the unknown malware by computing its similar known malware profile (Moshiri *et al.,* 2017). After detection, the next stage is pre-processing followed by training and testing the considered dataset. Testing process is carried by CTM (Cluster Tree Map) algorithm. CTM is used to cluster and classify the resilience data in the effective manner and providing the better result. The next stage is Knowledge inference. Inference is a database system method utilized to attack databases in which malicious users deduce sensitive information from complex databases at a higher stage. Finally, performance analysis is done and the results are validated. The flow chart of the proposed methodology is given in Figure 1.

Attacks exploiting Machine Learning Systems: Situational awareness for Cyber Defenses incorporates 7 fundamental viewpoints: "Monitoring a. current circumstance, effect of attack, how circumstances develop, performer conduct, why and how the present circumstance is caused, quality and how conceivable futures of the present circumstance." Based on this definition, it very well may be said that situational awareness gives the user the closer view and brief assessment of system.

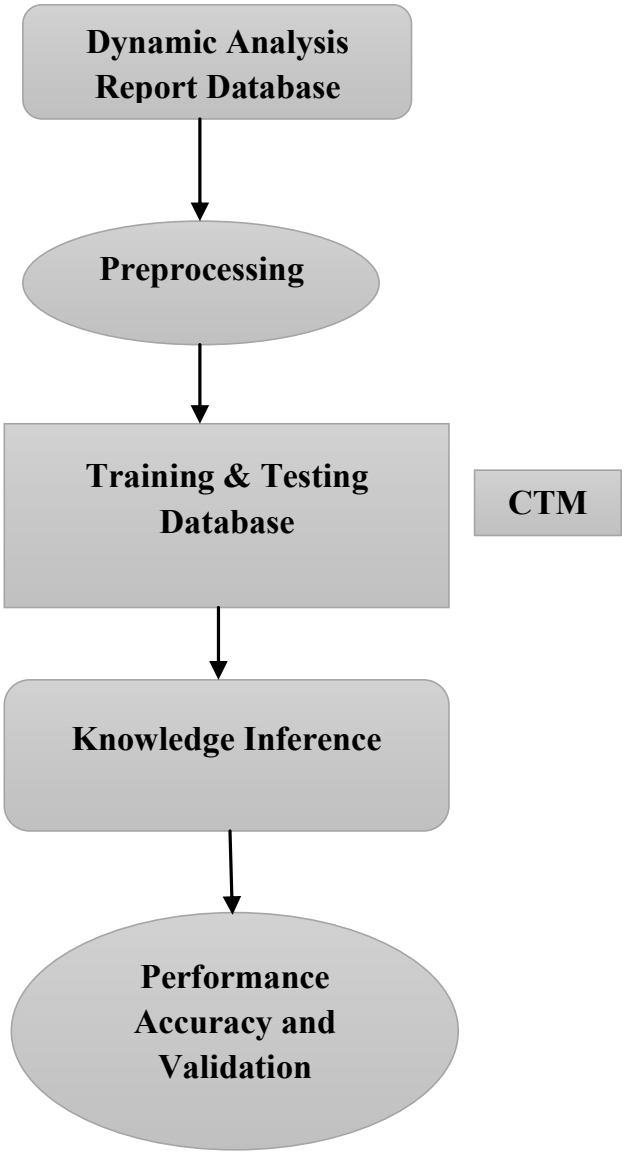

Figure 1. Flow chart of Proposed Methodology

Table 1. Weighted Average of calculated results

	TP rate	FP rate	Precisio-n	Recall	F-Measure	ROC area	Class
	0.998	0.001	0.999	0.998	0.999	0.999	cluster0
	0.999	0.002	0.998	0.999	0.998	0.999	cluster1
Weighted average	0.998	0.002	0.998	0.998	0.998	0.999	

Table 2. Proposed Calculation results

Correctly Classified Instances (5204)	99.85%
Incorrectly Classified Instances (8)	0.15%
Kappa statistic	0.9969
Mean absolute error	0.0017
Root mean squared error	0.0357
Relative absolute error	0.33%
Root relative squared error	7.12%
Total number of instances	5212
Algorithm Processing Time(Milliseconds):	4061

Table 3. Sensitivity and Specificity of Proposed and Existing System

	TP	TN	FP	FN	Sensitivity	Specificity
Existing	2434	2675	54	47	0.981	0.983
Proposed	2736	2468	5	3	0.999	0.998

Table 1. Weighted Average of calculated results

	TP rate	FP rate	Precisio-n	Recall	F-Measure	ROC area	Class
	0.998	0.001	0.999	0.998	0.999	0.999	cluster0
	0.999	0.002	0.998	0.999	0.998	0.999	cluster1
Weighted average	0.998	0.002	0.998	0.998	0.998	0.999	

Table 2. Proposed Calculation results

Correctly Classified Instances (5204)	99.85%
Incorrectly Classified Instances (8)	0.15%
Kappa statistic	0.9969
Mean absolute error	0.0017
Root mean squared error	0.0357
Relative absolute error	0.33%
Root relative squared error	7.12%
Total number of instances	5212
Algorithm Processing Time(Milliseconds):	4061

Table 3. Sensitivity and Specificity of Proposed and Existing System

	TP	TN	FP	FN	Sensitivity	Specificity
Existing	2434	2675	54	47	0.981	0.983
Proposed	2736	2468	5	3	0.999	0.998

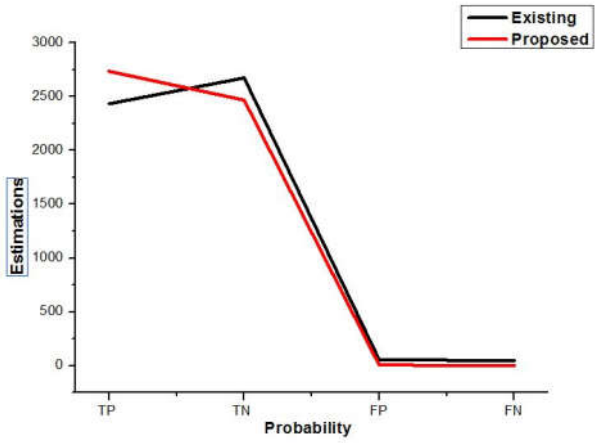

Figure 3. Graph for probability comparison

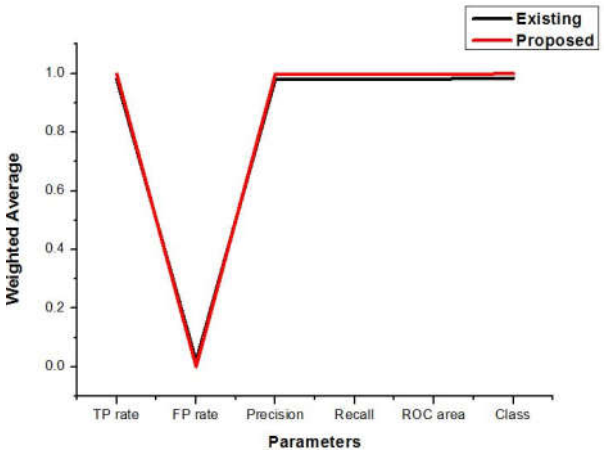

Figure 4. Graph for parameters comparison

Also, recognizing factors won't be easy to the point that even a few associations don't have awareness about their cyber-missions. Also, they have neither any adequate following components in their systems nor sensors which will lead them for attacks. Under these restrictions, it is extremely hard to have a powerful situational awareness. Best process is recognizing the basic missions which the association has; then implementing the tracking mechanisms, lastly characterizing proactive solutions for the security of system. A few examinations have been accomplished in this field. While a portion of the investigations (Tadda, 2010; Cumiford, 2006) by and large depict the term of situational awareness and standard strategies and applications which have been utilized so as to look after it; others center around more explicit techniques, for example, real-time multistage attack awareness and mission-centric cyber situational awareness (Mathew *et al.,* 2005; Jajodia *et al.,* 2011). Attack trees assume a huge process in indicating framework security and system as far as powerlessness and hazard recognizable proof (Ray, 2005). They are able to be mapped in different structures. Generally, while nodes overcome attacks, the root node is the global objective of the attacker which can likewise portrayed as an event.

Child nodes are the enhancements of this objective and branches are the way in which attacker can't be refined any longer. Every path in attack tree denotes one of a kind attack. In addition, attack trees can likewise be arranged literarily rather than graphically. In textual form, the 'AND' and 'OR' disintegrations are utilized and the outcomes of accomplish sub goals were displayed by them. Attack examples can be characterized so as to build the reasonableness of attack trees age and reuse (Moore, 2001). Attack design is the mapping of various kinds of attacks that incorporates a. the objective of the predetermined attack, b. the preconditions for utilize, c. the means for rehearsing attack, d. post conditions which are valid if the attack is made effectively (Moore, 2001). The preconditions contain suspicions which are connected with the normal practices of attacker and the qualities of the attack. The abilities, assets, access and information can be given for instance to preconditions (Moore, 2001). Then again, post conditions cover the picked up benefits when the attack was come about effectively. The main aim of using attack trees is to find out which types of attacks may be experienced because of critical vulnerabilities and to identify ways of attacks by using assets in the network. Moreover, with the help of attack trees the risks will be calculated and the precautions against

Table 4. Validation table

Parameters	TP Rate		FP Rate		Precision		Recall		F-Measure		ROC Area		Class	
Algorithms	Existing	Proposed	Existing	Proposed	Existing	Proposed	Existing	Proposed	Existing	Proposed	Existing	Proposed	Existing	Proposed
Label 1	0.978	0.998	0.017	0.001	0.981	0.999	0.978	0.998	0.98	0.999	0.984	0.999	Benign	cluster0
Label 2	0.983	0.999	0.022	0.002	0.98	0.998	0.983	0.999	0.981	0.998	0.984	0.999	Malignant	cluster1
Weighted Avg.	0.981	0.998	0.02	0.002	0.981	0.998	0.981	0.998	0.981	0.998	0.984	0.999		

Table 5. Comparison of parameters

Parameters	Existing		Proposed	
Correctly Classified Instances	5109	0.980614	5204	0.99847
Incorrectly Classified Instances	101	0.019386	8	0.00154
Kappa statistic		0.9611		0.9969
Mean absolute error		0.0264		0.0017
Root mean squared error		0.1354		0.0357
Relative absolute error		0.052693		0.00331
Root relative squared error		0.269474		0.07116
Total Number of Instances		5210		5212
Ignored Class Unknown Instances	2		0	
Algorithm Processing Time(Milliseconds):		36926		4061

attacks will be analysed and prioritized. In this way, possible risks are identified and arranged in an order based on their risk scores. While creating attack trees, web ontology language (OWL) is used for discovering the steps (attack tree nodes) and transitions -by using vulnerability- between steps. OWL interpretation requires huge memory and processing power. To overcome this performance bottleneck, attack trees are distributed in parallel fashion. Moreover, instead of creating attack tree by adding the devices one-by-one, the devices which have the same level of accessibility are united and their vulnerabilities are integrated. It is also assumed that possible attacks can affect all clustered devices. In this way, attack trees for very large networks can be processed faster. The infrastructure of distributed simulation and attack tree gives the possibility of using vulnerabilities coming from local vulnerability database. In attack tree, the visualization is based on vulnerabilities. An attack tree node includes a. the device b. vulnerability which is the source of an attack to this device (including pre/post conditions), c. gained privilege as a result of this attack on this device. Afterwards, by using attack trees, impact scores of the vulnerabilities and criticality levels, risk analysis is made and risk scores are calculated for each path in an attack tree. In the appendix, the sample of attack tree can be seen. (Goodall, 2008). An explanative taxonomy of various attacks whose objective is exploiting machine learning systems are: (a) Causative attacks changing the training process; (b) Attacks on integrity and availability, making false positives as a breach into a system; (c) Exploratory attacks exploiting the prevailing vulnerabilities; (d) Targeted attacks towards a certain input; (e) Indiscriminate attacks in which inputs will not work out. First type is a defense against exploratory attacks, in which an attacker can create an evaluation distribution that the learner predicts poorly. For defending against this attack, the defender can limit the access to the training procedure and data, making it tougher for an attacker to use reverse engineering. Also, the more difficult hypothesis space is, the tougher for an attacker to deduce the learned hypothesis. Additionally, a defender can limit the feedback provided to an attacker so that it becomes tougher to break into the system. Second type is a defense against causative attacks, in which an attacker can progress both evaluation and training distributed allocations.

Cluster Tree Mapping: In order to know about the data and in need of group data points to understand their collective behaviour, clustering is one of the go-to methods.Clustering techniques can group attributes into a few similar segments where data within each group is similar to each other and distinctive across groups. It is an unsupervised learning process finding logical relationships and patterns from the structure of the data. It is used for cases that involve:

- Discovering the underlying rules that collectively define a cluster
- Partitioning
- Discovering the internal structure of the data

Cluster Tree Map Algorithm

FeaturesSelect (Dataset)

```
Begin
//load dataset
source ← new DataSource(Dataset);
dataset ← source.getDataSet();
//create FeaturesSelection object
filter ← new FeaturesSelection();
//create evaluator and search algorithm objects
eval ← new CfsSubsetEval();
search ← new GreedyStepwise();
//set the algorithm to search backward
search.setSearchBackwards(true);
//set the filter to use the evaluator and search algorithm
filter.setEvaluator(eval);
filter.setSearch(search);
//specify the dataset
filter.setInputFormat(dataset);
//apply
newData ← Filter.useFilter(dataset, filter);
//save
f←new File(fp);
ff←f.getName();
driveLetter ← ff.split("\\.")(0);
fn←"attrs_"+driveLetter+".arff";
fns←"E:/input/"+fn;
saver ← new ArffSaver();
saver.setInstances(newData);
saver.setFile(new File(fns));
saver.writeBatch();
FilteredFeaturesDataset←fns;
End
```

J48 Algorithm: Classification is the process of building a model of classes from a set of records that contain class labels. Decision Tree Algorithm is to find out the way the attributes-vector behaves for a number of instances. Also on the bases of the training instances the classes for the newly generated instances are being found. This algorithm generates the rules for the prediction of the target variable. With the help of J48 algorithm the critical distribution of the data is easily understandable

Input: R the records of the dataset, the training data T, the attributes_available for computing the next branch

Output: J48 decision tree
Method:
Step 1: Node N is created.
Step 2: If all records in T have same target class
Step 3: Return N as a leaf node with target class.
Step 4: If attributes_available is empty
Step 5: Return N as leaf node with maximum target class for the records.
Step 6: Get best_attribute (T, attributes_available).
Step 7: attributes_available = attributes_available – best_attribute.
Step 8: Split the records based on best_attribute(best_attribute, T) //for each split, grown a subtree by calling the //Build J48 Decision Tree function
Step 9: For each split Ti of T on best_attribute
Step 10: Attach a new node returned by build J48 DecisionTree (split records Ti, attributes_available)
Step 11: End for
Step 12: End function

CTM_Test (Filtered Features Dataset)

```
Begin
datafile ← readDataFile(fpp);
validation←null;
data ← new Instances(datafile);
data.setClassIndex(data.numFeaturess() - 1);
// Do 10-split cross validation
split()() ← crossValidationSplit(data, 10);
// Separate split into training and testing arrays
trainingSplits() ← split(0);
testingSplits() ← split(1);
// Use a set of classifiers
ctm←newj48();
// Run for each model
// Collect every group of predictions for current model in a FastVector
predictions ← new FastVector();
// For each training-testing split pair, train and test the classifier
for i < trainingSplits.length
Begin
validation ← classify(mlprbf, trainingSplits(i), testingSplits(i));
predictions.appendElements(validation.predictions());
End
// Calculate overall accuracy of current classifier on all splits
accuracy ← calculateAccuracy(predictions);
// Print current classifier's name and accuracy in a complicated, but nice-looking way.
validation.toClassDetailsString();
validation.toMatrixString();
validation.toSummaryString();
new GenerateROC(fpp);
visualizetree(fpp);
End
```

In the login Page, the user is asked to enter username and password. After logging in, in background page values are validated using adminlogin.java. If entered information is valid, then the page is redirected to Adminpage.jsp otherwise it shows error then it is redirected to Login Page. Then Benchmark dataset which was collected from CLAMP datasets is uploaded to demonstrate the effectiveness of Malware Prediction. One can also view the dataset at this stage. Total number instances/records and total number of instances/ record attributes can be known. Decision Tree ROC visualization by threshold curve is obtained. An ROC curve is a commonly used way to visualize the performance of a machine learning classifier, meaning a classifier with possible output classes.A cluster tree is formed. Uploaded Dataset values are viewed and then pre-processing is done to select the best features among them. About 14 best features are selected from 70 features to consume the buffer memory to proceed further. Total number instances/records and total number of instances/ record attributes are obtained.

RESULTS AND DISCUSSION

In our proposed work, an independent method on the involvement of content experts to evaluate feature

manipulation costs has been applied and followed by resilience estimation. Then, unfavourable resilient classifiers are constructed. Numeric techniques depending on the preferability of experts has been found to make them aware of any such method which cancorrectly model the innovativeness and cyber attackers' skill.This unsupervised learning model is done using Cluster Tree Map (CTM) and J48 Algorithm. The CTM algorithm is applied to know the behaviour of group data points and to select and filter the appropriate features. J48 Algorithm has been formulated for classification and to find the behaviour of attributes. The true positive and false positive rate, precision, recall F-Measure, ROC area and class values of cluster 0 and cluster 1 are estimated. Form table 1, it is observed that the weighted average of the true positive and false positive rate, precision, recall F-Measure, ROC are 0.998, 0.002, 0.998, 0.998, 0.998 and 0.999 respectively. From the calculation results of table 2, it is observed that the Correctly Classified Instances (5204) is 99.85 %, Incorrectly Classified Instances (8) is 0.15%, Kappa Statistics is 99.69%, Mean Absolute error is 0.17%, Root mean squared error is 3.57%, Relative absolute error is 0.33%, Root relative squared error is 7.12%.

The total number of instances is about 5212. The time required for the algorithm to process is 4061 milliseconds. In table 3, Sensitivity and specificity for both the existing and proposed system are calculated from the TP, TN, FP, and FN values. Sensitivity values of existing and proposed systems are 0.981 and 0.999 respectively. It is observed that sensitivity of proposed system is greater than the existing system. The obtained Specificity values for existing and proposed system are 0.983 and 0.998 respectively. Proposed system shows greater specificity than existing system. From the graph shown in figure 3, it is observed that proposed methodology shows better probability than the existing method. The parameters like TP rate, FP rate, precision, recall, F-Measure, ROC area, Class are estimated and compared with the existing method. The weighted average values of label 1 and label 2 for all the above mentioned parameters were found to be better compared to the existing methodology and it is shown in table 4. The graph shown in figure 4 depicts the comparison of weighted average of both the existing and proposed methodology. From the table 5, it can be observed that correctly classified instances (0.99847), incorrectly classified instances (0.00154), Kappa statistic (0.9969), mean absolute error (0.0017), root mean squared error (0.0357), relative absolute error (0.00331), root relative squared error (0.07116), total number of instances (5212), ignored class unknown instances (0), algorithm processing time (4061) estimations are much better in proposed methodology than existing methodology.

Conclusion

CTM (Cluster Tree Map) has been used to cluster and classify the resilience data in the effective manner. The proposed methodology provided better results when all data types in the area of Cyber Network data analytics are focussed. Better accuracy is achieved by CTM algorithm. Feature manipulation costs are estimated. To make an independent approach on the involvement of content experts to estimate feature manipulation costs using CTM Algorithm. Thus, an independent availability on resilience evaluation is made and adversary resilient classifiers are constructed. Unsupervised learning model is utilized to create an awareness of any such method capable of correctly modelling the creativeness and skill of cyber attackers. In future, Resilience of attacks can be done by advanced classifiers rather than Machine Learning Classifiers for achieving better results.

REFERENCES

Alpert, C.J., Kahng, A.B., Yao, S.Z. 1999. Spectral partitioning with multiple eigenvectors. Discret. Appl. Math. 90, 3–26. (CrossRef)

Barreno M. *et al.,* 2010. "The security of machine learning", Journal Machine Learning, Vol. 81, Issue 2, pp. 121-148, November.

Buczak L. and Guven, E. 2016. ''A survey of data mining and machine learning methods for cyber security intrusion detection,'' IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153– 1176, 2nd Quart.

Buczak, A. L., & Guven, E. 2016. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176.

Chung, F. 1997. Spectral Graph Theory; American Mathematical Society: Providence, RI, USA.

Cumiford, L. D. 2006. "Situation Awareness for Cyber Defense," Information for the Defense Community.

Dubey, P. 2005. Recognition, mining and synthesis moves computers to the era of tera. Intel Tech. Magazine, 9(2):1–10.

Ford, V., & Siraj, A. 2014. Applications of Machine Learning in Cyber Security. In Proceedings of the 27th International Conference on Computer Applications in Industry and Engineering.

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G. and Vázquez, E. 2009. "Anomaly-based network intrusion detection: Techniques, systems and challenges," Comput. Secur., vol. 28, no. 1, pp. 18–28.

Girvan, M., Newman, M.E. 2002. Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA, 99, 7821–7826. (CrossRef) (PubMed)

Goodall, J. R. 2008. Introduction to visualization for computer security. In *VizSEC 2007* (pp. 1-17). Springer, Berlin, Heidelberg.

Jajodia, S., Noel, S., Kalapa, P. and Albanese, M. 2011. "Cauldron missioncentric cyber situational awareness with defense in depth," in Proc. Military Communications Conference, pp. 1339-1344.

Mathew, S., Britt, D., Giomundo, R. and Upadhyaya, S. 2005. "Real-Time Multistage Attack Awareness Through Enhanced Intrusion Alert Clustering," in Proc. Military Communications Conference, pp. 1801-1806.

Moore, A. P., Ellison, R. J. and Linger, R. C. 2001. "Attack Modeling for Information Security and Survivability," Carnegie Mellon Software Engineering Institute.

Moshiri, E., Abdullah, A. B., Mahmood, R. A. B. R., & Muda, Z. 2017. Malware Classification Framework for Dynamic Analysis using Information Theory. Indian Journal of Science and Technology, 10(21).

Nguyen T. T. T. and Armitage, G. 2008. "A survey of techniques for internet traffic classification using machine learning," IEEE Commun. Surv. Tuts., vol. 10, no. 4, pp. 56–76, Fourth Quart.

Pervez M. S. and. Farid, D. M. 2014. ''Feature selection and intrusion classification in NSL-KDD CUP 99 dataset employing SVMs,'' in Proc. 8th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA), pp. 1–6.

Ray I. and Poolsapassit, N. 2005. "Using Attack Trees to Identify Malicious Attacks from Authorized Insiders," in Proc. Computer Security - ESORICS 2005 Lecture Notes in Computer Science, v. 3679, pp. 231-246.

Revathi S. and Malathi, A. 2013. ''A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection,'' in Proc. Int. J. Eng. Res. Technol., pp. 1848–1853.

Sahoo, D., Liu, C.and Hoi. S. C. H. 2017. ''Malicious URL detection using machine learning: A survey.'' (Online). Available: https://arxiv.org/abs/1701.07179

Shi, J., Malik, J. 2000. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22, 888–905.

Simon, A., & Singh, M. 2015. An Overview of M Learning and its Ap. International Journal of Electrical Sciences Electrical Sciences & Engineering (IJESE), 22.

Tadda, G. P., Salerno, J. S. 2010. "Overview of Cyber Situation Awareness," in Cyber Situational Awareness Issues and Research, vol. 46, Springer, pp. 15-35.

Venkataramani, S., Raghunathan, A., Liu, J., & Shoaib, M. 2015. Scalable-effort classifiers for energy-efficient machine learning. In Proceedings of the 52nd Annual Design Automation Conference (p. 67). ACM.

Wu and S. X., Banzhaf, W. 2010. "The use of computational intelligence in intrusion detection systems: A review," Appl. Soft Comput., vol. 10, no. 1, pp. 1–35.

****