

A Machine Learning Algorithm Based on Inverse Problems for Cyber Anomaly Detection

Abstract

With the advancements in technology, humans have achieved undistinguished feats in 21st century which were thought to be beyond the reach previously. Digital communications have become an inseparable part of each sector. Healthcare, Education, E-Commerce, and Government profusely make use of the digital world. But, as these technologies have alleviated human efforts, it has also generated severe threats to cyber security. Digital communications are used at its' peak and it has given rise to cyber security issues too. In actual scenario, adversarial attacks are hard to detect with conventional methods as these cyber activities look very similar to genuine ones. There are many problems in anomaly and misuse detection of cybersecurity which can be considered as an inverse problem. In this paper, we have modeled anomaly detection system, Inverse Machine Learning Algorithm (IMLA), based on an inverse model approach with Riesz kernel and applying software system development concepts at each phase. The proposed approach IMLA have been compared with various supervised learning models. The experiments show the effectiveness of the proposed model.

Keywords: *Anomaly Detection; Inverse problems; Machine Learning; Cyber Analytics; Data Mining*

1. Introduction:

The aim of this paper is to present an efficient approach for anomaly detection in cybersecurity. It is a crucial part for any cybersecurity application because information security personal have to decide if a system is safe or not at any given moment. The high margin of accuracy in the operational cybersecurity is very necessary and quite beneficial for both the users and security personal. This paper analyses intrusion detection evaluation data set and provides a framework which can discriminate between good cybersecurity operations and the bad ones. This project focuses on the use of knowledge discovery methods with soft computing method to find a solution for this problem. A Hybrid classifier is proposed and used to analyze the data set and extract the decision rules from it. These rules enable the cyber security managers to make effective decisions and classify the operations as good or bad. The objective of the model developed here is to maximize *the security and minimize the risk on the behalf of system*.

Adversarial attack is a prevalent term for malicious entity using or involving in a cyber communication as interfering with or preventing normal cyber activities [1]. The intention can be to obtain unauthorized access to any cyber medium to compromise the confidentiality, integrity, and availability of an information system. According to the World economic

Forum, the rate of cybercrime had increased by 21 percent in 2017, which used to be holding steady during the mid-2000s. Around 46% of Americans have been victim to cybercrime in the past years. The Nilson Report estimates that in 2021, losses topped \$6 trillion. That represents a 62% increase over the previous five years.

In this paper, we have used DARPA 1999 intrusion detection evaluation dataset that was collected for a total of five weeks. The data for the first three weeks was assigned as the training set and the last two weeks was assigned as the testing dataset [2]. We have used Python due to its' efficient and professional structure for implementation of our algorithm. We have used Python libraries such as pandas, numpy, matplotlib for plotting the graphs, pickle etc. This work is divided into five modules such as Input module, Train module, Test module, Performance metrics module and Display module. We have implemented Inverse Machine Learning Algorithm (IMLA) and added some utility methods to measure the performance of the algorithms. The final output in the form of *Trained Model* can be hosted as a web service to further anomaly detections. Along with that xml diagrams such as Architecture diagram, Sequence diagram of the model and training modelling diagrams are also drawn.

2. Anomaly Detection

Anomaly detection is a challenging field of cybersecurity. It is a very *complicated* task to distinguish anomalies from normal actions. The aim of every method is to classify the anomalies with the normal behavior. It becomes very hard for humans to decide just by inspecting the operations with some detail.

The advancement in the information technology has information systems to electronically store all the details of actions. This led to the urgency of making a system which can automate access granting decision so that the stress on cybersecurity managers can be reduced and efficiency could be improved. There are lots of methods that have been proposed and covered in the literature that use supervised and unsupervised learning [3]-[10]. Most of these methods have used different classifying techniques with reasonable accuracy, but these studies not go beyond classification. As it plays a crucial role in the anomaly evaluation procedures so there must be a proper justification of the approval and rejection of a specific action and an automated system which can assist the cybersecurity professional in this process.

Figure 1 show the general approach that is followed while assessing the anomaly using mathematical modeling. The training dataset which consist of different features and attributes about an event, then they are fed to the network, after this the data is preprocessed. For example if some data value is missing or corrupted data is found it is removed to optimize the accuracy. In the next phase, a mathematical modeling technique is used for training the network and evaluating the anomalies and distinguishes between the normal and abnormal events to the cybersecurity manager.

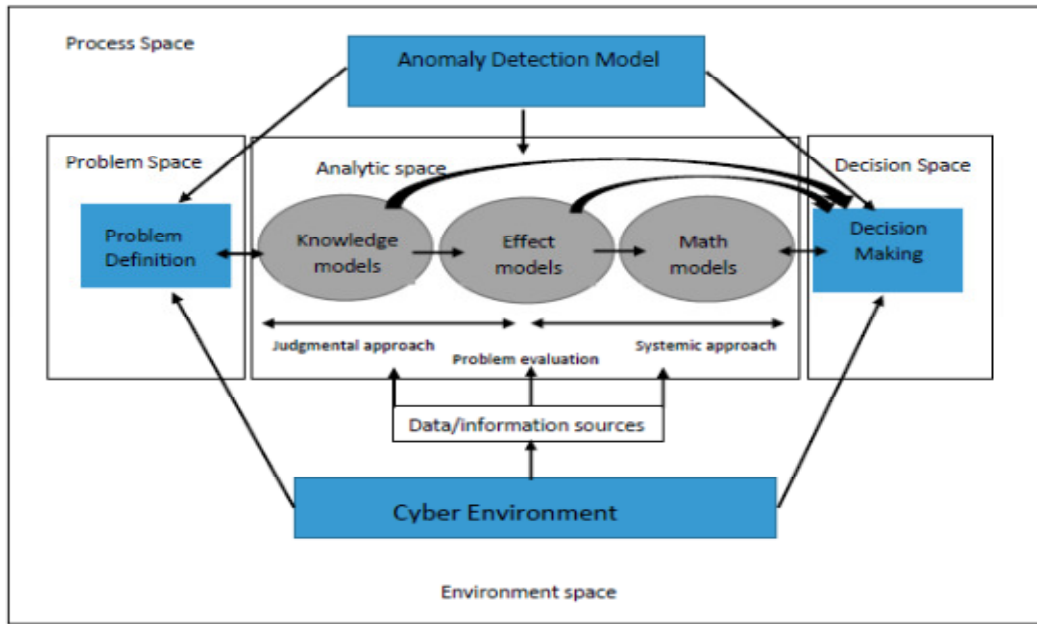


Figure 1. A general approach of anomaly evaluation using mathematical modeling

3. Model Analysis

Knowledge discovery is finding the patterns from the large dataset and extracting useful knowledge and information from it. It involves discovering patterns from the large datasets using different mathematical models. It is formally the intersection of Machine learning (ML) and artificial intelligence with statistics and data base system. There are different mathematical models available like neural networks, fuzzy, genetic algorithm, and support vectors along with their hybridizations that extract knowledge from the databases [11] – [13]. These soft computing methodologies are widely accepted for classification, clustering and predictions.

Many data properties estimation as knowledge discovery may be stated as a category of Inverse Problems. The category that we are introducing here is composed by problems, when we are interested in properties of an object, as in anomaly detection in cyber platform. We consider the problem of anomaly detection as an inverse problem and discuss how it can be solved by machine learning techniques.

In this paper, starting from a reformulation of the anomaly detection as an integral equation of inverse problems, we introduce an alternative machine learning algorithm derived by a well-known regularization method studied in [14]. Anomaly detection problem in Information Systems may be represented by an operator equation

$$Af = g \quad (2.1)$$

where A is a compact linear operator with a Riesz type kernel. Here f represents anomaly status of information system and g represents results of the experiments with the training data sets. The eq. (2.1) is a first kind of Fredholm integral equation with a Riesz type kernel. Since the integral operator has an unbounded inverse, the solution of integral equation is ill-posed.

Therefore a regularization technique is used to stabilize the solution by adding a new regularization parameter, stabilizer, and the equation (2.1) may be reformulated as a regularized inverse problem

$$(A^*A + \alpha I)f(\alpha) = A^*g \quad (2.2)$$

New stabilizer component α in the Eq. (2.2) will adjust input/output relationship. The integral Eq. (2.2) has an unknown function f which will be recovered from pre-processed data and will be used to detect anomalies.

The next step in the model to discretizing the Eq. (2.2) into matrix equation form. Then we show that this discrete reformulation can be employed to design a machine learning algorithm based upon a numerical solution of the equation (2.2).

The dataset which we used from Kaggle was originally collected and analyzed during a research collaboration of Worldline and the Machine Learning Group (<http://mlg.ulb.ac.be>) of ULB (Université Libre de Bruxelles) on big data mining and anomaly detection. Out of 284,807 records 492 were anomalies. This is a very unbalanced dataset. The dataset contains only numerical input variables which are a result of a PCA transformation. There are total 31 columns in the dataset, out of the first 28 columns entitled V1 to V28 are called Features, except the first column. The first column Time describes the seconds elapsed between each request and the last column which is Class has value 1 in case of anomaly and 0 otherwise.

We have implemented the IMLA algorithms of the anomaly detection using Python programming language. Python is very popular and professional to work with on machine learning projects. Python has particular tools like which are very useful in working with machine learning and we have used those tools in our analysis. The following Software and Hardware requirements were used in our computations: Windows 10, Python version 3.6, Anaconda version 3, Jupyter notebook/ipython, Python libraries (tensorflow, numpy, pandas, sklearn), RAM: 16 GB, Processor i7 and above.

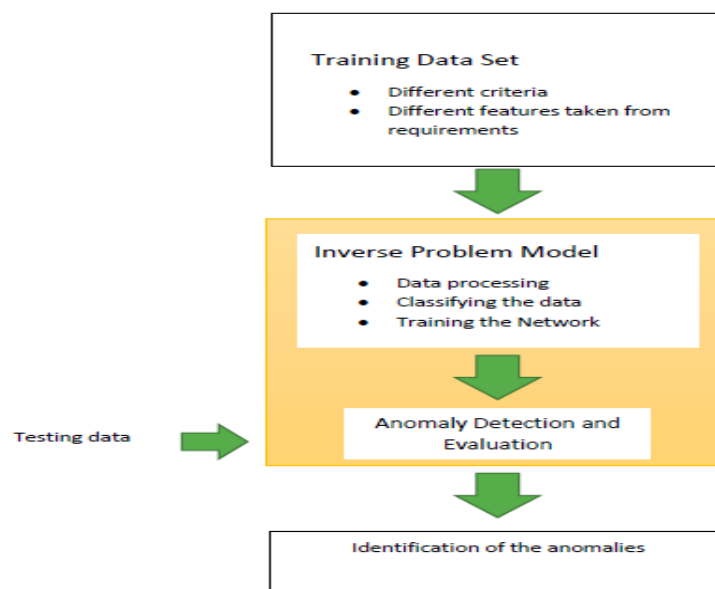


Figure 2. Core architecture of the system.

4. Results and Discussion

The process will start with getting the essential data sets for anomaly detection from authentic sources. For the analysis, 2 days of data from September 2013 is considered. Total 284,807 rows of data are considered.

This data is fed into the preprocessing module, where missing values and junk values of rows are removed. The data is also normalized (brought in the range of 0 to 1) to make the computations faster, this is an optional step.

The preprocessed data is then, randomly, split into train and test data and the output class is separated from them as well. The trained data is then fed into the classifier and the model is trained using this data as shown in Figure 3. Once the training is completed, the test data is used to validate the accuracy of the model. The model is retrained if the required accuracy is not obtained. Graphical representations of these accuracies are plotted to get a better visualization of the trained model. This trained model can be hosted as a web service to further detect anomalies. We have five modules, they are input module, train module, test module, performance metrics module, and display module. The proposed model IMLA have been compared with some of the recent relevant models SVM, Bayesian Network, and Decision Trees.

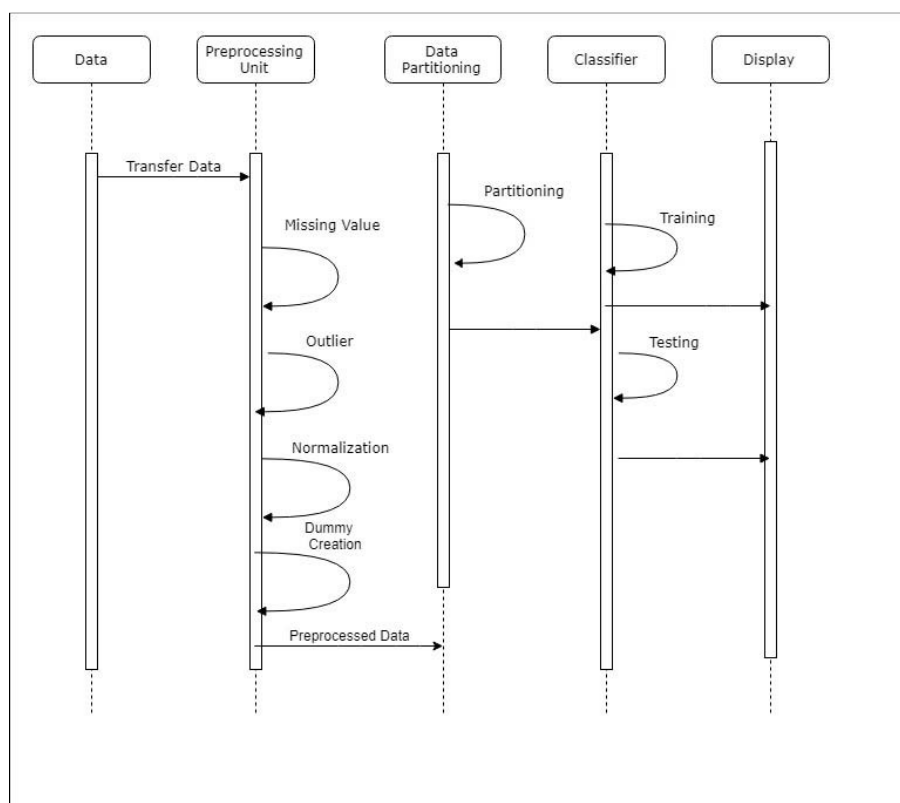


Figure 3. Sequence Diagram of the Model

The proposed model is designed and tested by following an object-oriented pattern and the correctness of the algorithms is checked by comparing the various evaluation metrics.

The resulting metrics and graphs of applying SVM, Decision Trees, Bayesian Network, and

IMLA algorithms on this dataset is provided to visualize and interpret them. As shown in Figure 4 – Figure 9, all the models give a high accuracy but to be not blindfolded by the accuracy alone as this dataset is highly unbalanced we have to also look at the false positive count.

IMLA on complete dataset: The IMLA classifier was applied to the complete dataset and an accuracy of 97% was obtained, but the instances that actually matter are the instances which are actually anomalies.

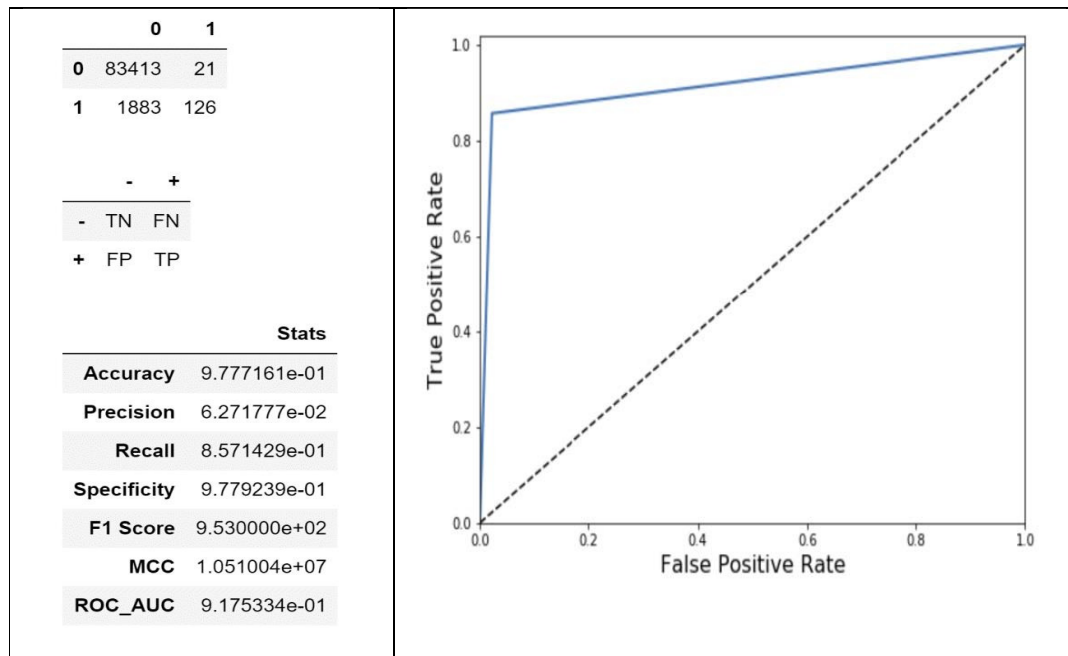


Figure 4. IMLA on complete dataset

As we can see from the False Positive count in the below confusion matrix 1883 instances were classified as normal actions where as in reality they were anomalies. Hence, this model is reasonable, but not great (Figure 4).

IMLA on Under-Sampled dataset: Since the IMLA did not work very well on the complete dataset because of the imbalanced nature of the dataset, let's apply the same on the Under-Sampled dataset.

As we can see we obtained a low false positive count and a good True Positive dataset and accuracy of 90%. So this model looks good. Let's compare this model with another Classifier i.e. Bayesian Network and see how it performs. (Figure 5).

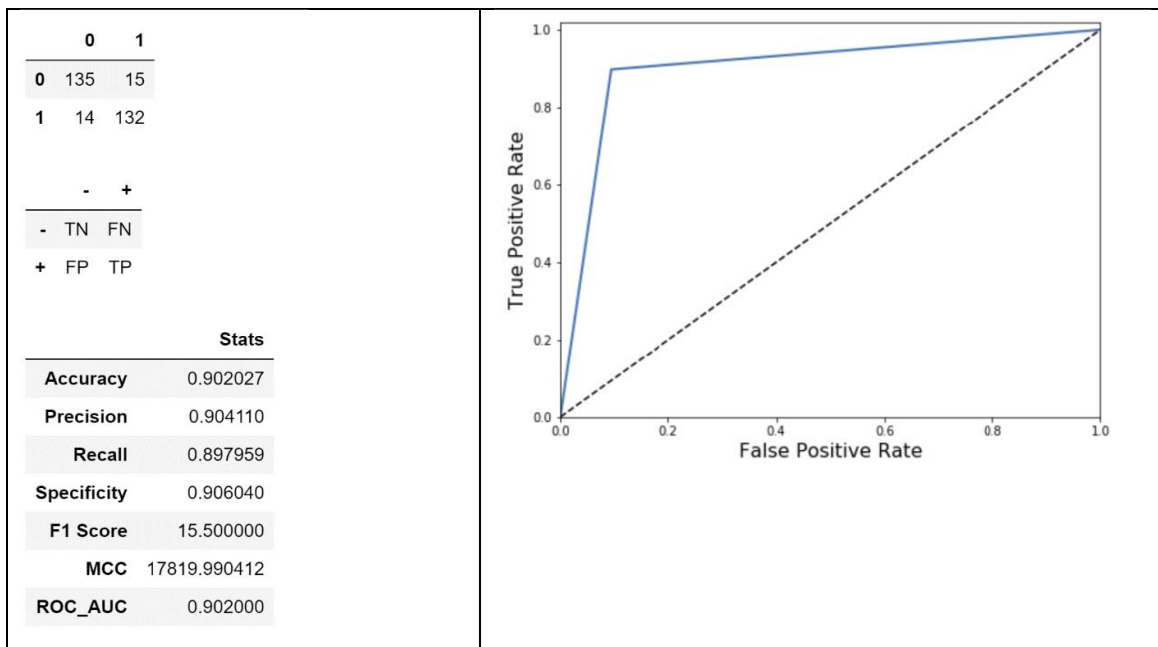


Figure 5. IMLA on Under-Sampled dataset

Decision Trees on Under-Sampled Data: Before moving on to a different classifier we applied Decision Trees from a standard sklearn library to check the correctness of the implemented algorithm.

As we can see from the confusion matrix and the stats table below, they are inline with those of the implemented version of it.

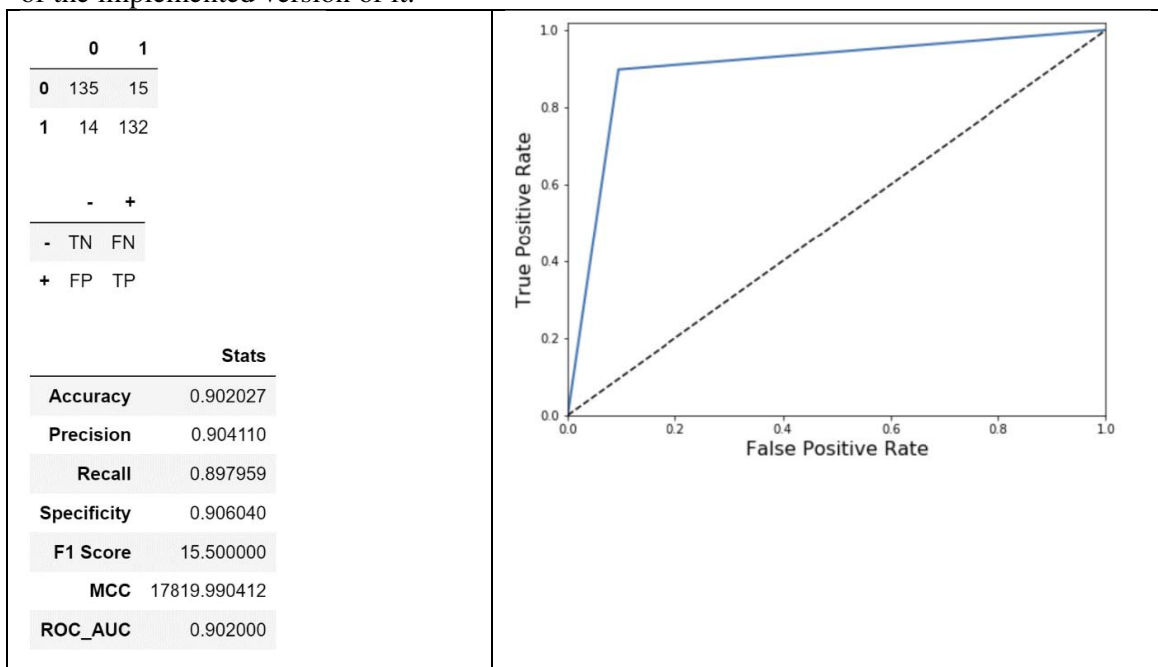


Figure 6. Decision Trees on Under-Sampled Data:

SVM on Original Data: The SVM classifier was applied to the complete dataset and a accuracy of 99% was obtained, as mentioned before the instances that actually matter are the instances which are actually anomalies.

As we can see from the False Positive count in the below confusion matrix only 15 instances where classified as normal actions where as in reality they were anomalies. So, this model performs pretty well on the unbalanced dataset as well.

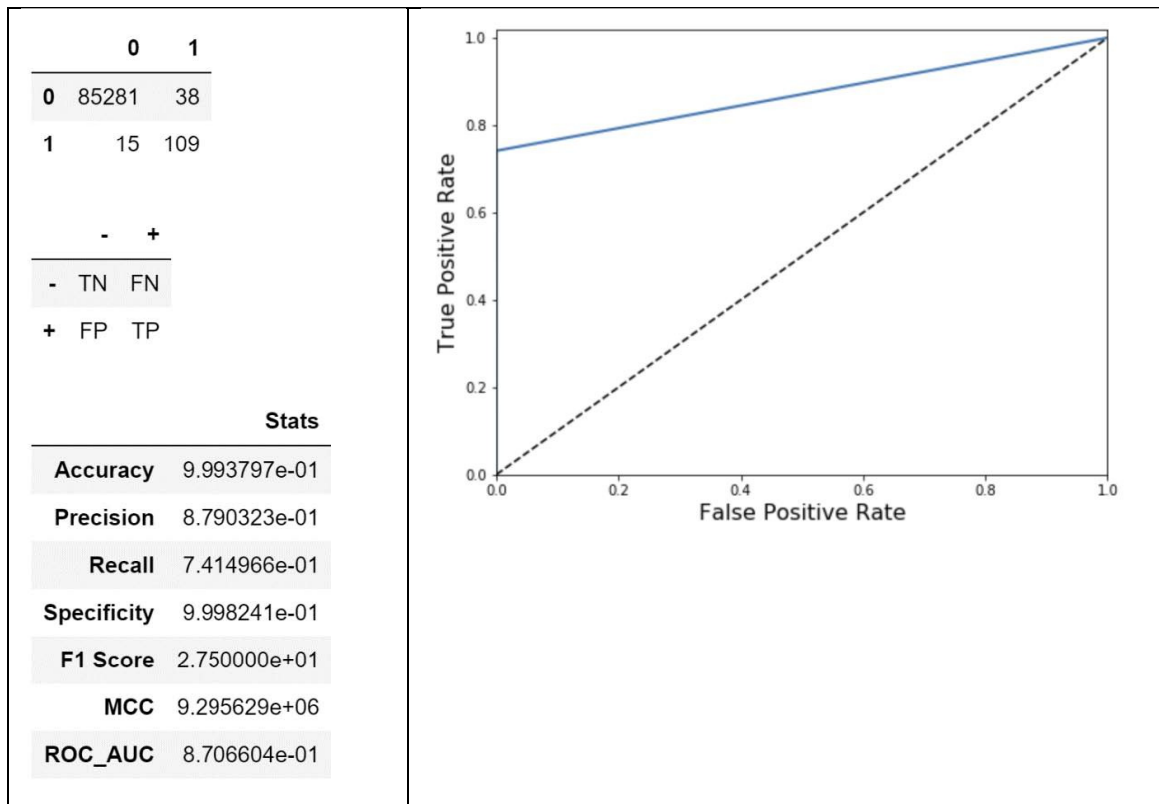


Figure 7. SVM on Original Data

SVM on Under-Sampled: Let's apply the same on the Under-Sampled dataset just to be double sure. As we can see we obtained a very low false positive count, a good True Positive count and accuracy of 90%. This model performs the same as IMLA for equally proportionate dataset.

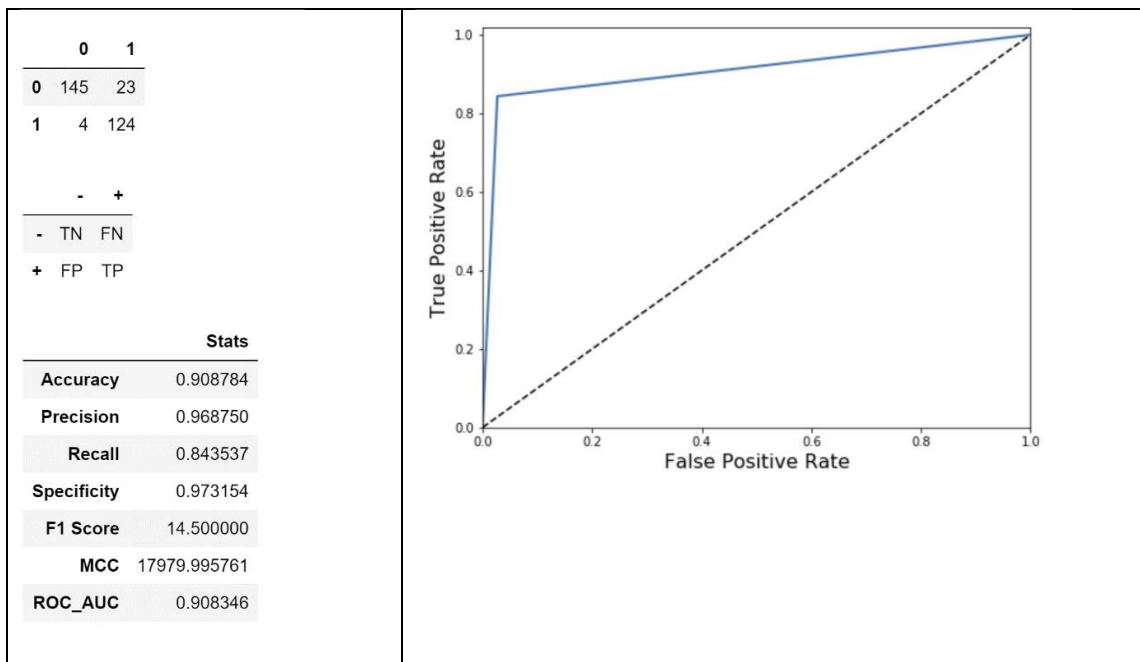


Figure 8. SVM on Under-Sampled

Bayesian Network using Sklearn Library: Before concluding we applied Bayesian from a standard sklearn library to check the correctness of the implemented algorithm. As we can see from the confusion matrix and the stats table below, they are in line with those of the implemented version of it.

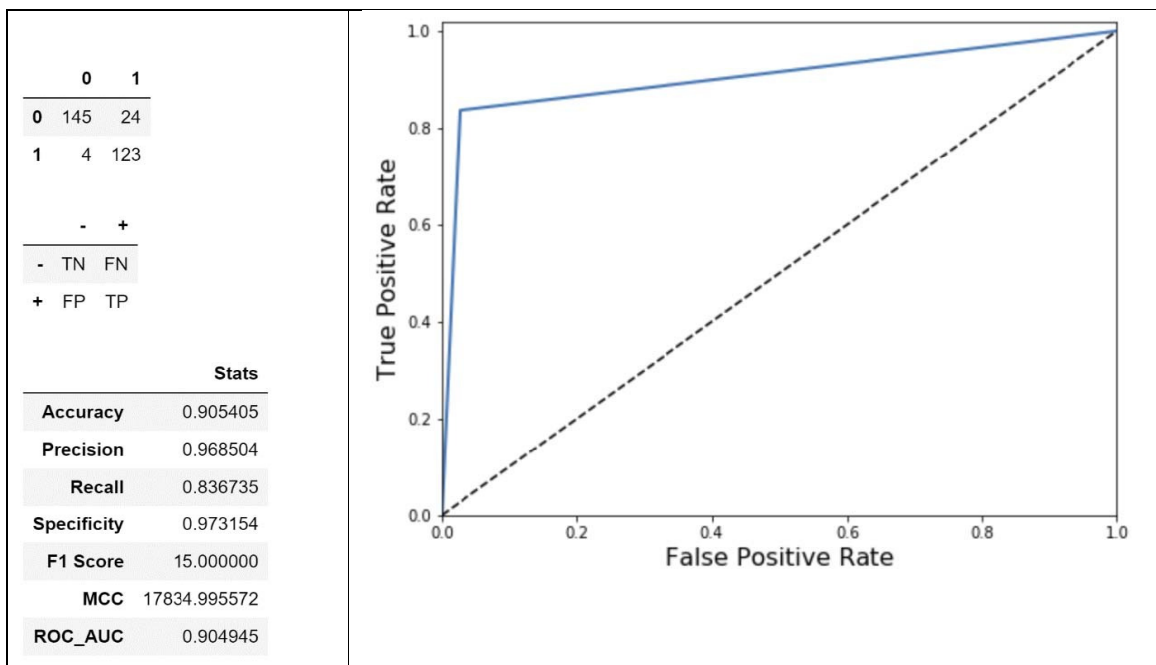


Figure 9. Bayesian Network using Sklearn Library

213 In Figure 10, we summarized all of our findings in a single diagram from Figure 4-9.

Output / Analysis

IMLA			Decision Trees			Bayesian Network			SVM		
Under-Sampled Data											
	0	1		0	1		0	1		0	1
0	135	15	0	135	15	0	145	24	0	145	23
1	14	132	1	14	132	1	4	123	1	4	124
Original Data											
	0	1								0	1
0	83413	21							0	85281	38
1	1883	126							1	15	109

214
215 **Figure 10. Results of experiments**

216 5. Conclusion:

217 The experiment demonstrates the execution and viability of our framework and exhibit the
218 usefulness of learning the anomalies of the cybersecurity events. Comparative investigations
219 reveal that the accuracy of the framework is more than **90%** percent over a wide variety of
220 the input information. The framework is additionally adaptable for taking care of extensive
221 volumes of events. All of the algorithms; SVM, Decision Trees, Bayesian Network, and
222 IMLA over complete data set and under sampled data shows the model trained on under
223 sampled data give high accuracy compared to the counterpart. The results of IMLA are also
224 close to accuracy calculated using the contemporary models which verify the performance of
225 our model. As future enhancement, a web-interface can be created to provide a front end to
226 load data files and display results. Overall, these results motivate further research of
227 additional empirical studies of its parameters sensitivity and scalability using reinforced
228 learning which enables the model to detect anomalies and train itself at the same time.

230 231 References

- 232 [1] S. Dua & X. Du, Data Mining and Machine Learning in Cybersecurity, *CRC Press*, (2011).
233
234 [2] W. H.Chen, S.H. Hsu, and S. H. Shen, Application of SVM and ANN for intrusion detection.
235 *Computers and Operations Research* 32(10), 2617–2634, 2015.
236
237 [3] M. Bhuyan, D. Bhattacharyya, J. Kalita, "Network anomaly detection: Methods systems and
238 tools", *IEEE Commun. Surv. Tuts.*, vol. 16, no. 1, pp. 303-336, 2014.
239
240 [4] S. X. Wu, W. Banzhaf, "The use of computational intelligence in intrusion detection systems:
241 A review", *Appl. Soft Comput.*, vol. 10, no. 1, pp. 1-35, 2010.
242

- 243 [5] IBM, Feb. 2015, [online] Available: <http://www.iss.net>.
- 244 [6] B. Morel, "Artificial intelligence and the future of cybersecurity", *Proc. 4th ACM Workshop*
- 245 *Secur. Artif. Intell.*, pp. 93-98, 2011.
- 246
- 247 [7] H. Brahmi, B. Imen, B. Sadok, "OMC-IDS: At the cross-roads of OLAP mining and intrusion
- 248 detection" in *Advances in Knowledge Discovery and Data Mining*, New York, NY, USA:
- 249 Springer, pp. 13-24, 2012.
- 250
- 251 [8] S. Benferhat, T. Kenaza, A. Mokhtari, "A Naïve Bayes approach for detecting coordinated
- 252 attacks", *Proc. 32nd Annu. IEEE Int. Comput. Software Appl. Conf.*, pp. 704-709, 2008.
- 253
- 254 [9] M. Blowers, J. Williams, "Machine learning applied to cyber operations" in *Network Science*
- 255 *and Cybersecurity*, New York, NY, USA:Springer, pp. 55-175, 2014.
- 256
- 257 [10] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, C. Kruegel, "2014 Exposure: A passive DNS
- 258 analysis service to detect and report malicious domains", *ACM Trans. Inf. Syst. Secur.*, vol. 16,
- 259 no. 4, Apr. 2014.
- 260
- 261 [11] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, C. Kruegel, "Disclosure: Detecting botnet
- 262 command and control servers through large-scale netflow analysis", *Proc. 28th Annu. Comput.*
- 263 *Secur. Appl. Conf. (ACSAC'12)*, pp. 129-138, Dec. 2012.
- 264
- 265 [12] S. Khan, "Rule-based network intrusion detection using genetic algorithms", *Int. J. Comput.*
- 266 *Appl.*, vol. 18, no. 8, pp. 26-29, Mar. 2011.
- 267
- 268 [13] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, K. Dai, "An efficient intrusion detection system based
- 269 on support vector machines and gradually feature removal method", *Expert Syst. Appl.*, vol. 39,
- 270 no. 1, pp. 424-430, 2012.
- 271
- 272 [14] A. Sever, Neural Network Algorithm to Pattern Recognition in Inverse Problems, *Applied*
- 273 *Mathematics and Computation* 22(2013) 484–496.