Vilnius University
Institute of Data Science and Digital Technologies
L I T H U A N I A

INFORMATICS (09 P)

APPLYING MACHINE LEARNING METHODS FOR EARLY DETECTION OF CYBER SECURITY INCIDENTS

Viktoras Bulavas

October 2018

Technical Report DMSTI-DS-09P-18-1

VU Institute of Data Science and Digital Technologies, Akademijos str. 4, Vilnius
LT-08663, Lithuania
www.mii.lt

Abstract

There are numerous sources for network intrusion detection data: for example, network traffic, system host logs, user activity, such as mail or browsing, use of smart devices and similar. All this data comes in big volumes, velocity and variety. Analysis of such data is essential for making anomaly detection and intrusion prevention decisions. Common data processing steps, following the acquisition of data, are pre-processing, which helps to reduce the number of dimensions, and visualization, which helps observation of distinct features in real time. Both steps, further discussed in this paper are required for better understanding of contained intrusion phenomena, such as data theft, malware activity or hacking attempts. Visualization helps further understand data by elaborating the well-hidden data properties and features. Numerous methods of multi-dimensional data visualization are currently available to assist data scientist or information security analyst in the broad landscape of intrusion data analysis. For simplicity, visualization methods in this report are categorized as direct, linear projection, non-linear projection and other. In this paper, attention is drawn to linear projection, in particular principal components analysis, helping to select the most informative dimensions of the data. Principal Component analysis provide indication of anomalies in network. Decision Tree method is utilized to provide decision criteria for anomaly recognition as intrusion. Investigation in this research demonstrates, that combination of PCA and Decision Tree methods allows classification of anomalies such as Smurf, Satan, Neptune, Portsweep, Ppsweep with probabilities higher than 95% with depth of tree set to 4 and PCA components set to 10. Nevertheless, Nmap and Teardrop anomalies are classified purely, therefore future investigation of Decision Tree hyper parameters is needed to increase classification accuracy.

Keywords: network intrusion detection, decision trees, principal component analysis, visualization

Contents

1 Introduction

Data visualization is a graphical data presentation method used for better understanding of chosen data. Keim, Mansmann et al. [1] define visual analytics as an approach that combines interactive visualizations with automatic analysis methods for a more comprehensive perception, reasoning and decision making process when processing massive and complex datasets. Data dimension reduction and visualization techniques help identify and evaluate the structure of data, abnormalities and similarities. Information security analysts' work requires visual presentation of network intrusion data for rapid detection of anomalies. Data visualization methods are assisting information security data scientist or information security analyst to get better understanding of the big data. Dzemyda, Kurasova and Zilinskas [2] classify visualization methods into direct visualization and projection visualization methods:

1. Direct visualization methods, when features of a multi-dimensional object are presented in a certain visual form. Using these methods, the selected dimensions of data are presented in a visual form on a two dimensional plane. Direct visualization methods can be further classified as geometric, symbolic and hierarchical.

2. Linear and nonlinear projection methods help to present multidimensional objects in a smaller number of dimensions of space (also known as dimension reduction methods). Linear projection visualization methods can be further classified into Principal Component Analysis, Linear Discriminant Analysis and Projection Pursuit. Non-linear projection methods can be further classified into Multi-dimensional Scaling, Locally Linear Embedding, Isometric Feature Mapping, Principal Curves [2].

In this paper (see Sections IV and V), attention is drawn to linear projection, in particular Principal Components Analysis as introduced by Hotelling [3], helping to select the most informative dimensions for intrusion detection.

This paper is organized as follows. Section II introduces intrusion detection problem and some related data source issues. Section III introduces related work of other authors in visualization methods, used for a broader context of network data, and machine learning methods for intrusion detection in particular. Section IV discusses Principal Component Analysis, used in pre-processing and visualization of intrusion detection data, in Section V, results of experiment with intrusion detection data and visualization of Principal Components, supported with Decision Tree method are presented, and finally in Section VI conclusions and directions for future research are drawn.

2 Intrusion detection problem and related data issues

2.1 Intrusion detection problem

Intrusion detection, signalling malicious activity or policy violations at network or system level, is a well-recognised problem of cybersecurity domain. Current network Intrusion Detection Systems (IDS) utilize three main technics: anomaly detection, misuse detection and hybrid of anomaly and misuse detection. Misuse detection systems use signatures that describe already known attacks. Such systems require regular update of rules. Such rule update is often acquired from a vendor, providing cloud update service. On the other hand, anomaly detection systems maintain normal data model and are capable of detecting deviation. Anomaly detection was originally introduced by Anderson [4] and Denning [5]. The main advantage of anomaly detection algorithms is that these algorithms detect new forms of attacks, because these attacks deviate from the known behaviour [5].

Machine learning based anomaly detection requires supervision and regular specialist reviews due to currently still high false positive rate of detecting previously unseen, but normal system behaviours. With an increasing frequency of cyber-attacks, reviews take more and more time of cyber security specialists, which is a challenge. This indicates highly demanded area for research aiming to increase threat detection accuracy and training speed.

2.2 Datasets and related issues

Primary sources [6] of intrusion detection data, further defined as datasets, are network flows from other network domains and local network, enriched with host-based user behaviour and system level content, as shown in Fig. 1, which is needed to detect anomalous behaviour and various types of intrusion attacks.

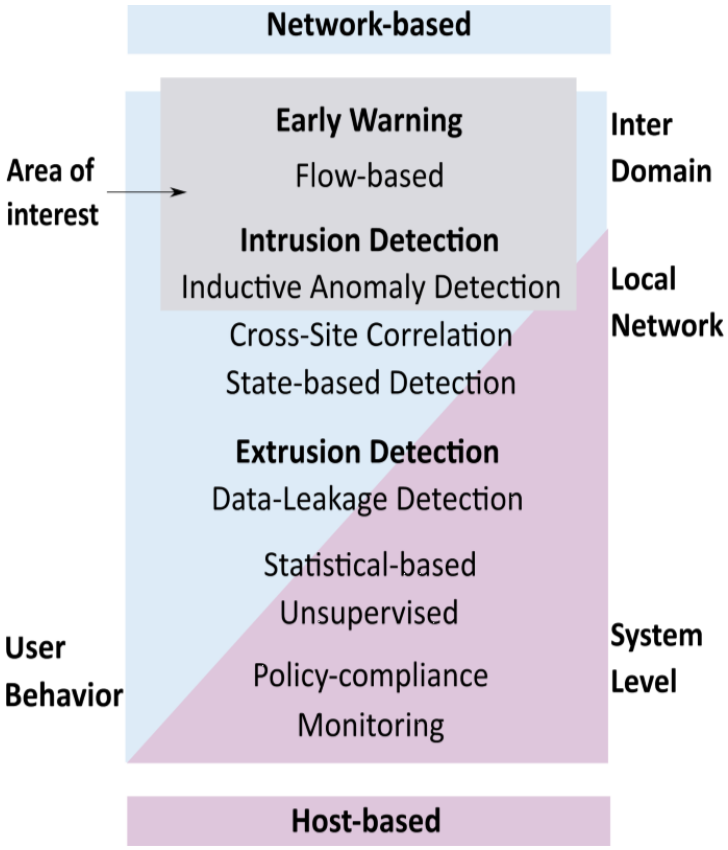

Fig. 1. Layers of intrusion and extrusion detection data [6].

The router or switch can collect IP network traffic as it enters and exits the interface. Flow monitoring has become a prerequisite for monitoring traffic in networks. A network flow is predominantly defined as a unidirectional sequence of packets that share the exact same packet attributes: ingress interface, source IP address, destination IP address, IP protocol, source port, destination port, and type of service.

The host has the ability to generate system level and user behaviour data, usually not obtainable directly from network flows, but related on a temporal axis. Such data would be for example failed login attempts. All this data has temporal dimension, which is needed for real life observation of intrusion.

Security analysts can identify many attacks from network traffic data, falling into a few generic types. S. Hettich and S. D. Bay, in their DARPA [7] analysis [8] define the following classes of malicious user attacks:

- Denial of Service (DoS), which is an attempt to deny aimed users computing or network resources,

- User to Root (U2R), granting root access to the attacker,

- Remote to Local (R2L), granting local network access to the attacker,

- Probe (or Scan), collecting information about the network resources.

There are many real datasets, generated from real world network traces, available to the research community, that have been used by the researchers to evaluate the performance of their proposed intrusion detection and intrusion prevention approaches. Far from being complete, the list includes: DARPA 1998 [9] and 1999 traces by Lincoln Laboratory, USA, KDD'99 [7], [8], CAIDA [10] datasets by University of California, USA, The Internet Traffic Archive and LBNL traces by Lawrence Berkeley National Laboratory, USA [11], DEFCON by the Shmoo Group, USA [12], ISCX IDS 2012 and CIC IDS 2017 [13] by University of Brunswick, Canada, and others.

One of most easily accessible for research and education purposes intrusion detection datasets is KDD'99, generated for KDD Cup Contest of 1999. It has been updated as NSL–KDD and made available for download at University of Brunswick, Canada. The NSL–KDD dataset consists of 41 dimensions [14].

NSL–KDD attributes can be classified into four different classes as discussed below. The first and second types of traffic features are time-based:

1) Basic features: attributes that can be extracted from a TCP/IP connection (ingress interface, source IP address, destination IP address, IP protocol, source port, destination port, and IP type of service).

2) Host features: examine only the connections in the past 2 seconds that have the same destination host as the current connection, and calculate statistics related to protocol behaviour, service, etc.

3) Service features: examine only the connections in the past 2 seconds that have the same service as the current connection. However, now popular slow probing attacks scan the hosts using a time interval as defined by botnet control centre.

4) Content features: unlike most of the DoS and Probing attacks, the R2L and U2R attacks don't have a similar sequential pattern. The R2L and U2R attacks are embedded in the data portions of the packets, and normally represent only a single connection. To detect such attacks, IDS needs specific features in the data portion to recognise as an anomaly, for example a number of failed login attempts. These features are called content features.

Problem of NSL-KDD data set is limited number of attack types. CIC IDS 2017 [13] introduces more types of attacks such as DDoS, Brute Force, XSS, SQL Injection, Infiltration, and Botnet. The dataset is completely labelled and more than 80 network traffic features are extracted and calculated.

2.3 Time spread of intruder attacks

The basic style attack types, occurring in cyber space have been tested with different types of Machine Learning algorithm ensembles to reduce the false alert rate (FAR), which varies in a range of 5% depending on the method used and the type of attack. Meanwhile multiple new threats are employing diverse multi-layer attack vectors, and the intruder reconnaissance is spread in time, therefore requiring behaviour-based

detection techniques. Anomaly goes unnoticed during the reconnaissance, and is often discovered after the incident occurs. New threats are not labelled until the incident is researched, and labels are not readily available for training.

2.4 Change of network flows formats

Even though NetFlows may be still the most frequent due to Cisco's popularity in the networking industry, other network equipment vendors provide similar network flow monitoring technology, which implies, that training has to be tailored for the specific flow record type. Furthermore, standards are updated, and new evolve, like the Internet Protocol Flow Information eXport (IPFIX), therefore intrusion detection systems will have to be updated.

3 Related Work

In this section a review of machine learning and visualisation methods (direct network awareness, dynamic and port mapping visualisation) is presented.

3.1 Direct network awareness visualization methods

There are many examples of applying visualization to improve network monitoring and intrusion detection. Lakkaraju et al. [15] visualize flows using three levels of granularity: a galaxy view, which shows the whole network, a small multiples view, which shows the information for a selected set of hosts, and a view which shows the behaviour of one machine. Similarly, Yin et al. [16] use parallel coordinates to monitor the state of a network.

3.2 Temporal network dynamics visualization methods

To represent data dynamics in temporal dimension, Musa and Parish [17] have used animation. Among those using spatial layouts of time, McPherson, Ma et al. [18] visualize port activity with time on the y-axis while Abdullah, Lee et al. [19] visualize alarms from an IDS for a large IP space in columns where the x-axis of each column is time. Livnat, Agutter et al.[20] describe polar layouts with time on the radius, while Keim, Mansmann et al. [1] use an angular measure for time.

3.3 Port mapping visualization methods

Another widely used visual technique is port-based visualization, because port activity of a network is essential for port scanning. Goodall, Lutters et al. [21] map each IP address to a row to produce a timeline of activity. Connections between IP addresses are drawn as lines between rows. Fischer and Keim [22] combined some interactive visualization views, with a tree map to display the most active ports and node-link graphs to represent and examine inner connections between different ports of network hosts. Port based approaches can show the most active ports during a time-period, but the port scan patterns are not obvious.

Kim and Reddy [23] demonstrated, that each sample of network flow data could be represented as an image frame or a video stream. For example, the image may represent traffic volume in bytes or packets going to a destination or the traffic between a source and destination pair.

Multiple pieces of data can be represented as different colours of an image leading to clear visual presentation and simpler analysis.

However, all these methods are effective only after proper selection of dimensions of analysis for intrusion detection.

3.4 Machine learning methods used for intrusion detection

Various authors proposed an ensemble of machine learning models as a probable way for solving intrusion detection problems.

3.5 Learning methods used for intrusion detection

Various authors proposed an ensemble of machine learning models as a probable way for solving intrusion detection problems. Ensembles of methods could be bagging, boosting or bootstrapping.

As shown in Fig. 2, bagging of ensembles of ML methods train combinations of base machine learning models.

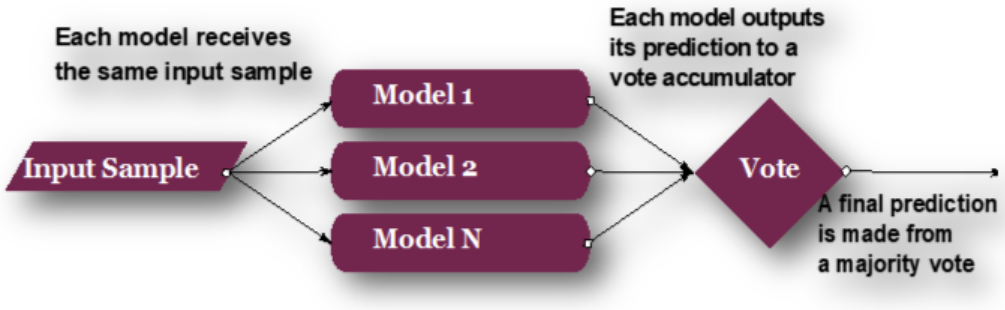

Fig. 2. Ensemble of ML Methods.

Ensembles were demonstrated to be an efficient way of improving predictive accuracy and/or decomposing a complex, difficult learning problem into some easier tasks. Krawczyk, Minku, Gama, Stefanowski and Wozniak [24] have introduced a taxonomy of ensembles, used for data stream analysis, as presented in Fig. 3:

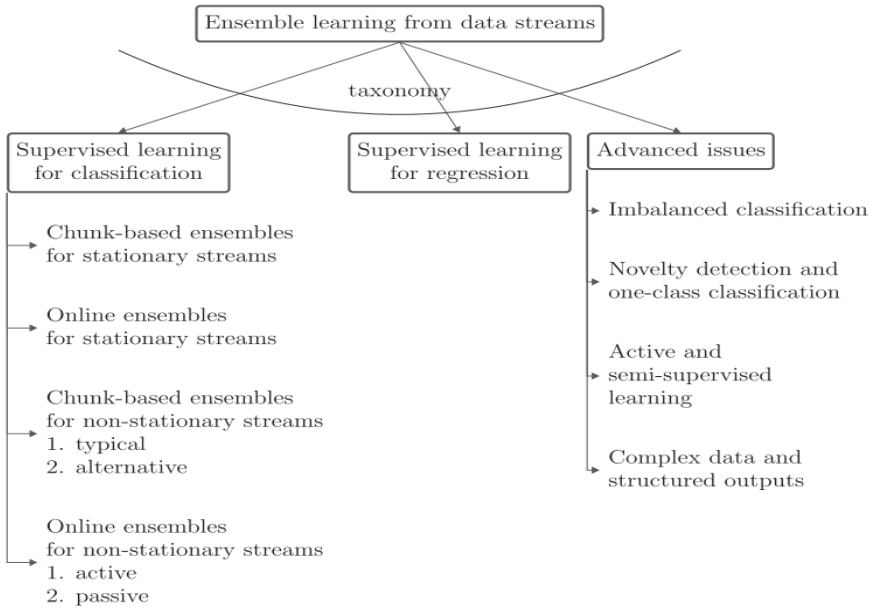

Fig. 3. Taxonomy of Ensemble learning from data streams [24].

Machine learning approaches used in intrusion detection include Decision Trees, Inductive Learning, Naive Bayes, Random Forest, Artificial Neural Networks, Fuzzy

Systems, Evolutionary Computation, Artificial Immune Systems, Hidden Markov, Sequential Pattern Mining, Swarm Intelligence [25], [26] and other.

According to the Wolpert's "no free lunch" theorem [27], there is not a single classifier that is appropriate for all the tasks, since each algorithm has its own domain of competence. However, Chen et al. [28] have demonstrated, that flexible neural tree (FNT) model can be used to reduce the number of features, when used for intrusion detection task. Dimension reduction will be further discussed to assist in reducing the complexity, associated with intrusion detection data.

There are several known ensemble machine learning issues, such as the number and types of base models to use, the combining method to use, and how to maintain diversity among the base models. Current IDS require an immense amount of data to learn, and data from one source (or location) is not enough. Experts are concerned with a need of constant retraining for IDS and refeeding same data into different models of an Ensemble.

4 Principal Component Analysis in intrusion detection

The common definition of Principal Component Analysis (PCA) was introduced by Hotelling [3]. It is said, that for a set of observed vectors $\{u_i\}$, i $\in \{1,..., N\}$, where N is number of vectors, the q principal axes $\{E_j\}, j \in \{1,\ldots q\}$ are those orthogonal axes onto which the retained variance under projection is maximal. It can be shown that the vectors E_j are given by the q dominant eigenvectors of the covariance matrix C of vector v, such that eigenvectors E_j and corresponding eigenvalues λ_i are solution to $CE_i = \lambda_i E_j$ equation. The vector $v_i = E^T(u_i - \bar{u})$, where $E = (E_1 \ldots E_q)$, is thus a q-dimensional reduced representation of the observed vector u_i.

PCA is one of the most successfully used feature extraction methods for traffic analysis. Brauckhoff, Salamatian and May [29] discuss implementing PCA method for anomaly detection and issue of right number of Principal Components for analysis. Ringberg et al. [30] discusses the sensitivity of PCA for anomaly detection, issues related to number of Principal Components, impact of anomaly size and gives a comprehensive study of the related issues on Abilene and Geant networks. Issariyapat and Kensuke [31] discuss using PCA for MAWI network and using the information from packet header for detecting anomaly.

Keerthi and Surendiran [32] carried out experiments with PCA using various classifier algorithms on two benchmark intrusion detection datasets namely, NSL–KDD and UNB ISCX 2017. Their experiments show that the first 10 Principal Components are effective for classification. The classification accuracy for 10 Principal Components is about 99.7% and 98.8%, nearly same as the accuracy obtained using original 41 features for KDD and 28 features for ISCX, respectively.

Amiri [33], Olusola [34], Zargari [35] and others, based on PCA analysis, proposed methods of selecting the best representing data features for intrusion detection: Service, Source bytes, Destination bytes and Destination host error rate. These features explain about 97% of variance. The remaining 37 features explain up to 99,7%, and 80 network features predict 99,97% of attacks.

5 Experiment

The objective of experiment in this research was to visualise different types of attack data, available in the NSL-KDD dataset. Table I introduces attacks and their types, available in NSL–KDD data.

TABLE I. Attack Types in NSL–KDD

No.	Name	Type
1.	Back	DoS
2.	buffer_overflow	u2r
3.	ftp_write	r2l
4.	guess_passwd	r2l
5.	imap	r2l
6.	ipsweep	probe
7.	land	DoS
8.	loadmodule	u2r
9.	multihop	r2l
10.	neptune	DoS
11.	Nmap	probe
12.	Perl	u2r
13.	Phf	r2l
14.	Pod	DoS
15.	portsweep	probe
16.	rootkit	u2r
17.	satan	probe
18.	smurf	DoS
19.	Spy	r2l
20.	teardrop	DoS
21.	warezclient	r2l
22.	warezmaster	r2l

To reduce computer resources needed for this experiment, amount of data loaded for analysis was limited to 20% (NSL-KDD-master 20 percent training set.csv). Column labels were added in order to facilitate import into the open-source data visualization, machine learning and data mining toolkit Orange 3 [36]. Further, visualization experiment, applying Principal Component Analysis and Decision Tree was accomplished.

A value of 10 Principal Components parameter was chosen, as recommended by Keerthi and Surendiran [32]. A simple Orange 3 workflow (see Fig. 4) is used for visualization, which gives best visibility with Components PC1 and PC2, see Fig. 5.

Fig. 4. Principal Component Analysis workflow using Orange 3 software.

However only a few attack types are visually separable in Fig. 5: Neptune (DoS), Satan (Probe), Nmap (Probe) and Portsweep (Probe), when PC1 is greater than 1.16.

Further in the experiment (see Fig. 6), selections of different Principal Component pairs were investigated, with the second best option provided by the pair PC1 and PC3, unfortunately other pairs gave little visual insight.

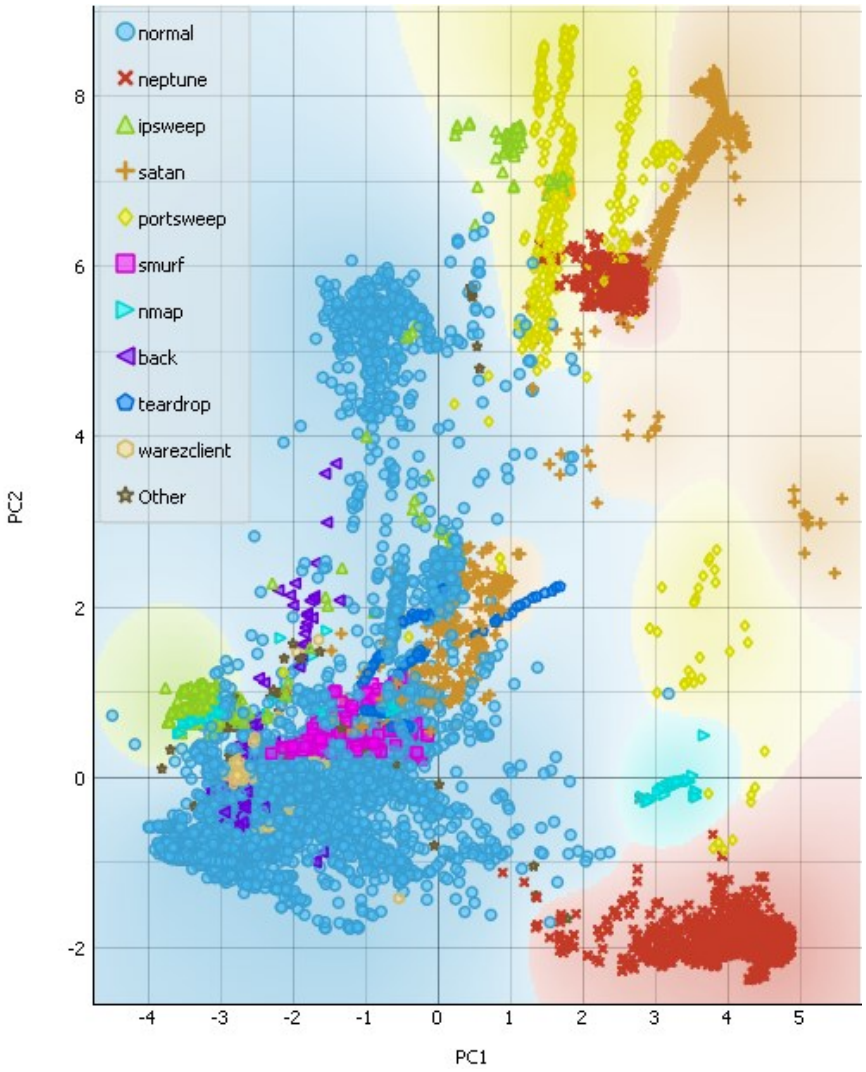

Fig. 5. Principal Component PC1-PC2 Analysis for NSL-KDD data using Orange 3 software.

In Fig. 6 it is shown, that PC3 brings not much additional visibility, as classes of attacks are not separated from normal traffic.

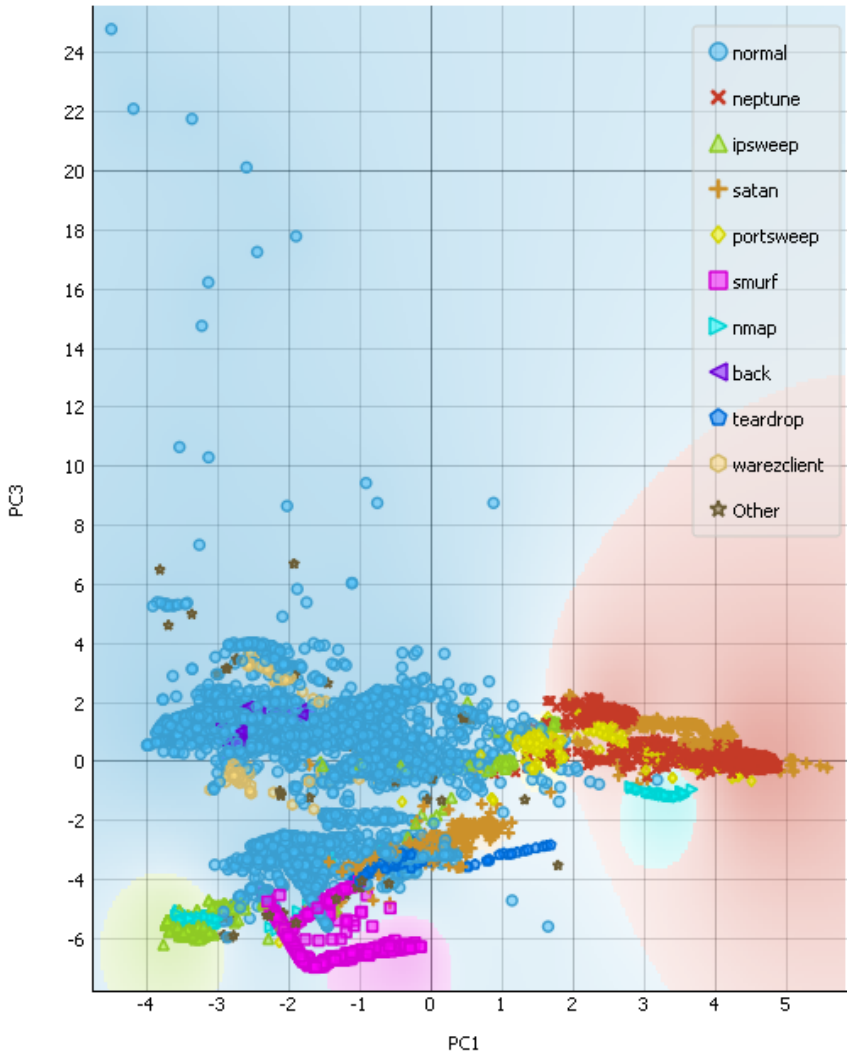

Fig.6. Principal Component PC1-PC3 Analysis for NSL_KDD data using Orange 3 software.

Solution is brought by Decision Tree analysis visualization method (see Fig. 7).

For the purpose of experiment reproducibility, related Orange workflow for PCA analysis is presented in Fig. 7.

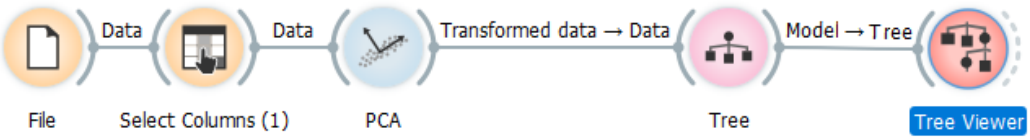

Fig.7. PCA and Decision Tree Analysis workflow using Orange 3 software.

In Fig. 8 the results of classification of NSL-KDD dataset Primary Components with Decision Tree are presented:

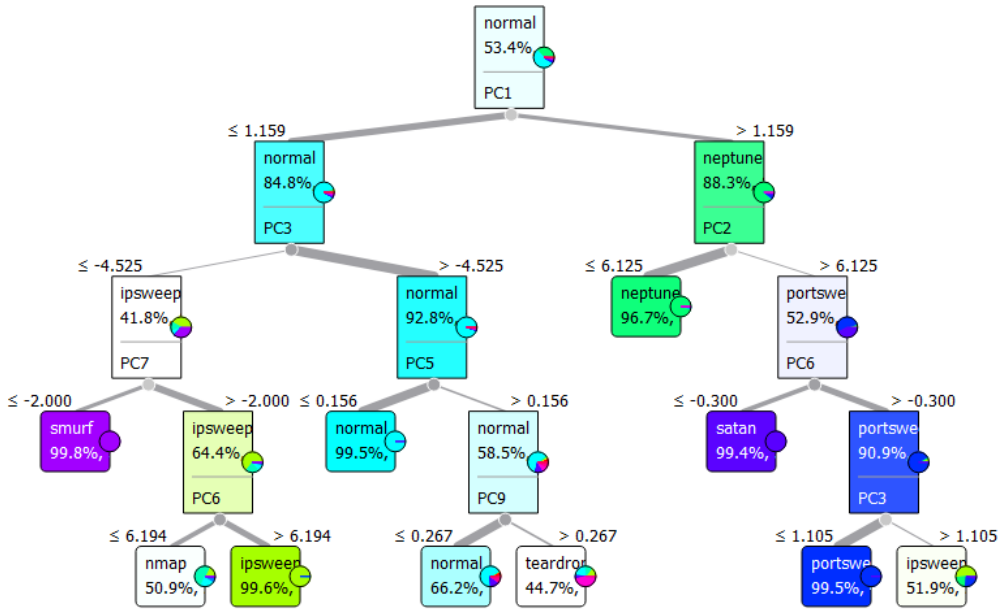

Fig.8. Decision Tree Analysis for NSL-KDD data using Orange 3 software.

By increasing the depth of the Decision Tree, one can distinguish other attacks, like Ipsweep (Probe), Smurf (Dos) and Teardrop (Dos), not previously clearly visible in PCA visualisations (Fig. 5 and 6).

Using more levels of Decision Tree, additional decision rules appear, explaining more intrusion types.

6 Conclusions

The resulting PCA analysis with Orange 3 software un-conceals the informative structure of dataset, however is not sufficient for visual decision making.

In the next step, using Decision Tree method it is possible to get human readable classification rules that are closely related to intrusion detection rules.

Investigation shows, that combination of PCA and Decision Tree methods allows classification of intrusions such as Smurf, Satan, Neptune, Portsweep, Ipsweep with probabilities higher than 95% with depth of tree set to 4 and PCA components set to 10. Nevertheless, Nmap, Teardrop intrusions are classified purely, therefore deeper Decision Tree is needed to increase classification accuracy.

Using more levels of Decision Tree, human readable rules can be further defined, bringing additional information for decision rules.

Future experiment and analysis could be performed using more detailed data source CIC IDS 2017 [13]. According to Sharafaldin, Lashkari, and Ghorbani, the abovementioned source, enriched with 80 network features, contains more than 28 informative principal components.

7 References

[1] D. a. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, “Challenges in Visual Data Analysis,” in Tenth International Conference on Information Visualisation (IV’06), 2006.

[2] G. Dzemyda, O. Kurasova, and J. Zilinskas, Multidimensional Data Visualization. Springer, 2012.

[3] H. Hotelling, “Analysis of a complex of statistical variables into principal components,” J. Educ. Psychol., vol. 24, no. 6, pp. 417–441, 1933.

[4] J. P. Anderson, “Computer security threat monitoring and surveillance,” 1980.

[5] D. E. Denning, “An Intrusion-Detection Model,” in 1986 IEEE Symposium on Security and Privacy, 1986, pp. 118–118.

[6] R. Koch, “Towards Next-Generation Intrusion Detection,” in 2011 3rd International Conference on Cyber Conflict, 2011, no. February, pp. 151–168.

[7] “KDD Cup 1999 Data,” 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[8] S. Hettich and S. D. Bay, “The UCI KDD Archive [http://kdd.ics.uci.edu],” Univ. California, Dep. Inf. Comput. Sci., 1999.

[9] R. P. Lippmann et al., “Evaluating intrusion detection systems without attacking your friends: The 1998 DARPA intrusion detection evaluation,” DARPA Inf. Surviv. Conf. Expo. 2000. DISCEX ’00. Proc., pp. 12–26 vol.2, 1999.

[10] The Cooperative Association for Internet Data Analysis, “CAIDA - The Cooperative Association for Internet Data Analysis,” CAIDA. 2010.

[11] Lawrence Berkeley National Laboratory, “The Internet Traffic Archive,” 2010. [Online]. Available: http://ita.ee.lbl.gov/index.html.

[12] The Shmoo Group, “Defcon,” 2011.

[13] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization,” in Proceedings of the 4th International Conference on Information Systems Security and Privacy, 2018, no. January, pp. 108–116.

[14] Y. Bouzida and F. Cuppens, “Efficient intrusion detection using principal component analysis,” Proc., 2004.

[15] K. Lakkaraju, W. Yurcik, and A. J. Lee, “013 NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness,” Proc. 2004 ACM Work. Vis. data Min. Comput. Secur. - VizSEC/DMSEC ’04, 2004.

[16] X. Y. X. Yin, W. Yurcik, Y. L. Y. Li, K. Lakkaraju, and C. Abad, “VisFlowConnect: providing security situational awareness by visualizing network traffic flows,” IEEE Int. Conf. Performance, Comput. Commun. 2004, 2004.

[17] S. Musa and D. J. Parish, “Visualising communication network security attacks,” in Proceedings of the International Conference on Information Visualisation, 2007.

[18] J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti, and M. Christensen, “PortVis: A Tool for Port-Based Detection of Security Events,” Proc. Int. Symp. Vis. Cyber Secur. - VizSec, p. 73, 2004.

[19] K. Abdullah, C. Lee, G. Conti, J. A. Copeland, and J. Stasko, “IDS RainStorm: Visualizing IDS alarms,” in IEEE Workshop on Visualization for Computer Security 2005, VizSEC 05, Proceedings, 2005.

[20] Y. Livnat, J. Agutter, S. Moon, R. F. Erbacher, and S. Foresti, “A visualization paradigm for network intrusion detection,” Proc. from 6th Annu. IEEE Syst. Man Cybern. Inf. Assur. Work. SMC 2005, 2005.

[21] J. R. Goodall, “Introduction to Visualization for Computer Security,” in VizSEC 2007, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–17.

[22] F. Fischer and D. a Keim, “VACS: Visual Analytics Suite for Cyber Security,” IEEE VAST Chall. 2013, 2013.

[23] S. S. Kim and A. L. N. Reddy, “A study of analyzing network traffic as images in real-time,” in Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., 2005, vol. 3, pp. 2056–2067.

[24] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, “Ensemble learning for data stream analysis: A survey,” Inf. Fusion, vol. 37, pp. 132–156, Sep. 2017.

[25] F. Gharibian and A. A. Ghorbani, “Comparative Study of Supervised Machine Learning Techniques for Intrusion Detection,” in Fifth Annual Conference on Communication Networks and Services Research (CNSR ’07), 2007.

[26] A. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Commun. Surv. Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.

[27] D. H. Wolpert, “The Supervised Learning No-Free Lunch Theorems,” Proc. 6th Online World Conf. Soft Comput. Ind. Appl., vol. 50 Suppl, pp. 25–42, 2001.

[28] Y. Chen, A. Abraham, and B. Yang, “Feature selection and classification using flexible neural tree,” Neurocomputing, vol. 70, no. 1–3, pp. 305–313, Dec. 2006.

[29] D. Brauckhoff, K. Salamatian, and M. May, “Applying PCA for Traffic Anomaly Detection: Problems and Solutions,” in IEEE INFOCOM 2009 - The 28th Conference on Computer Communications, 2009, pp. 2866–2870.

[30] H. Ringberg, A. Soule, J. Rexford, and C. Diot, “Sensitivity of PCA for traffic anomaly detection,” ACM SIGMETRICS Perform. Eval. Rev., vol. 35, no. 1, p. 109, Jun. 2007.

[31] C. Issariyapat and K. Fukuda, “Anomaly detection in IP networks with principal component analysis,” 2009 9th Int. Symp. Commun. Inf. Technol., pp. 1229–1234, 2009.

[32] K. Keerthi Vasan and B. Surendiran, “Dimensionality reduction using Principal Component Analysis for network intrusion detection,” Perspect. Sci., vol. 8, pp. 510–512, 2016.

[33] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, “Mutual information-based feature selection for intrusion detection systems,” J. Netw. Comput. Appl., vol. 34, no. 4, pp. 1184–1199, 2011.

[34] A. A. Olusola, A. S. Oladele, and D. O. Abosede, “Analysis of KDD & apos ; 99 Intrusion Detection Dataset for Selection of Relevance Features Analysis of KDD ’ 99 Intrusion Detection Dataset for Selection of Relevance Features,” vol. I, no. January, pp. 16–23, 2016.

[35] S. Zargari and D. Voorhis, “Feature selection in the corrected KDD-dataset,” in Proceedings - 3rd International Conference on Emerging Intelligent Data and Web Technologies, EIDWT 2012, 2012.

[36] J. Demšar et al., “Orange: Data Mining Toolbox in Python,” J. Mach. Learn. Res., vol. 14, pp. 2349–2353, 2013.