Insider Threat Detection Using Principal Component Analysis and Self-Organising Map

*Naghmeh Moradpoor
School of Computing (SoC)
Edinburgh Napier University
Edinburgh, United Kingdom
*n.moradpoor@napier.ac.uk

Martyn Brown
School of Computing (SoC)
Edinburgh Napier University
Edinburgh, United Kingdom
40135758@live.napier.ac.uk

Gordon Russell
School of Computing (SoC)
Edinburgh Napier University
Edinburgh, United Kingdom
g.russell@napier.ac.uk

ABSTRACT

An insider threat can take on many aspects. Some employees abuse their positions of trust by disrupting normal operations, while others export valuable or confidential data which can damage the employer's marketing position and reputation. In addition, some just lose their credentials which are then abused in their name. In this paper, we use Principal Component Analysis (PCA) in conjunction with Self-Organising Map (SOM) for insider threat detection within an organisation. The results show that using PCA before SOM increases the clustering accuracy.

CCS CONCEPTS

• Security and privacy → Intrusion/anomaly detection and malware mitigation → Intrusion detection systems

KEYWORDS

Insider Threat, Unsupervised Machine Learning, Self-Organising Map, Principal Component Analysis

1. INTRODUCTION

In terms of a definition, there is no one clear taxonomic statement about what makes something an insider threat. One strong definition comes from [11], where it defined an insider threat to be a past or present employee who uses current or past authorised access to a system to exceed or misuse that access to negatively affect confidentiality, integrity, or availability of an organization's systems. Although not recent, Bradley Manning is still frequently in the news, as the Army Private who leaked a treasure trove of classified military document to WikiLeaks in 2009 [10]. Here weaknesses in information control allowed a very junior employee access to an incredible amount of critical and private information, and even the identification of the employee in question relied on an informant rather than system-based technology. Data loss is in no way just historic, and there seems to be no sign that past issues are helping shore up systems against breach happening now. It was widely reported that in 2015, internal actors were responsible for 43% of all data loss [8]. Indeed, insider threats are seen as the main security threat in 2017. Recently 20 employees were arrested in China for illegally selling customer data, in a fraud said to be worth $7m [12]. The employees worked for Apple, and used their access to gather user's names, phone numbers, and Apple IDs. Although still early in the investigations, it would seem that the employees would have had legitimate access to the data, and simply abused that access for purely financial gains. According to a survey report by Haystax [13], 67% of insider attacks had appropriate credentials. Of these, 60% were managers. Contractors amounted to 57%, while regular employees amounted to 51%. The reasons why the insider attacks were carried out is also interesting. Haystax indicated 55% of attacks were focused on making money. In motivation terms, 42% cited sabotage while 38% said espionage [13]. The risks imposed by the insider threat are certainly significant, and so it is important that companies take appropriate steps to combat the threat. Preventing insider attacks actually happening has proved challenging, so the main research focus tend to focus on detecting and thus mitigating. Attack detection has taken many different approach, such anomaly detection of a user activity, role-based activity deviation, monitoring honeypots, and even measuring psychological factors.

In this paper, we use SOM which is a competitive and unsupervised machine learning algorithm for insider threat detection within an organisation. This is done by employing PCA first on the entire dataset in order to capture metrics such as: explained, latent, principal component coefficients and variance in each variable explained by each PC. The results from PCA helps us to give an insight into the underlying structure of the data in terms of its principal components rather. The results show improvements in SOM clustering accuracy after employing PCA. The remainder of this paper is organised as follows. In Sections 2, we review the related work on insider threat detection using machine learning follows by our six demo scenarios and entire dataset in Sections 3 and 4, respectively. This is trailed by the implementations and results in Section 5 and followed by conclusions in Section 6 and references.

2. RELATED WORK

Insider threat detection problems have been studied widely in current literature. This covers extensive categories such as: real time and non-real time, host-based & network-based user as well as use of machine learning/data mining approaches. Given that this paper focuses on the insider threat detection using machine learning algorithms, we briefly discuss related work on this for completeness. Authors in [1]

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

SIN '17, October 13–15, 2017, Jaipur, IN, India
© 2017 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-5303-8/17/10.
https://doi.org/10.1145/3136825.3136859

employed supervised Modified k-NN with Community Anomaly Detection System (CADS) & meta-CADS frameworks on user access logs. They presented that Modified k-NN outperforms k-NN. Authors in [5] used real time supervised Support Vector Machines (SVM) on a combination of human biological signals. They captured 90% detection accuracy with electroencephalography (EEG) which increased by 5% after adding the electrocardiogram (ECG). Authors in [2] employed unsupervised Ensemble- Graph- Based Anomaly Detection (E-GBAD) with Stream Mining and Graph Mining on system logs where each system log specifies by a token with eight attributes. They demonstrated that E-GBAD is more effective than traditional single-model approach. Authors in [3] used real-time unsupervised Deep Neural Networks (DNNs) & Recurrent Neural Networks (RNNs) with Tenserflow. They presented that DNNs and RNNs outperform PCA, SVM and Isolation Forest (IF). Authors in [7] employed unsupervised feature transformation PCA on their own developed dataset. Their system monitors user activities to construct features. At the end they employed PCA in order to reduce the generated features. Authors in [4] proposed real- time Anomaly Detection In Streaming Heterogeneity (RADISH) system where they used unsupervised k-NN and k-d tree. They presented that the k-d tree performs much faster than k-NN. Authors in [9] employed Unsupervised Modified IF and Supervised Random Forests (RF) algorithms where they obtained a ROC score of 0.77% for the unsupervised approach, and a classification accuracy of 73.4% for the supervised approach. In this paper, we use SOM in combination with PCA for insider threat detection to ascertain whether using PCA could improve the clustering accuracy of SOM.

3. DEMO SCENARIOS

In this paper, six demo scenarios have been identified by ZoneFox [6] to ensure familiarisation of the insider threats. ZoneFox is a market leader in User Behavior Analytics that helps businesses protect their critical data against the insider threats. The six demo scenarios include: three scenarios of: Data Theft, Endpoint Security Processing and Shadow IT Risk for permanent staff, one scenario of: Privileged User Data Breach for temporary staff and two scenarios of: Data Security and Protect Sensitive Folders for third party staff. In our demo scenarios, permanent staff is an individual who has been a member of the company's engineering/sales team. Temporary staff is a short-term member who has been employed for the busy period and a third party staff is an individual who has been employed to work with one of the client systems. The names are all fictional.

3.1 DEMO SECANRIO 1: DATA THEFT

For an organisation, *Data Theft* commonly includes stealing sensitive information related to its staff/clients in particular or its business in general. In Demo Scenario1, we focus on the threat of data theft from employees by following the Insider Threat Kill Chain with a particular focus on the one proposed by ZoneFox. In general, Insider Threat Kill Chain discusses how people within organisations can work together to help prevent insider risks before they become a problem. To achieve this, ZoneFox's Insider Threat Kill Chain model focuses on four groups of people within the organisation: 1) people who handed in their notice, 2) people who look around file server, 3) people who download backup software and 4) people who copy zip files to removable devices. For example, in Demo Scenario 1, Charlotte who is a member of the company's engineering team backs up files to a removable disk drive.

3.2 DEMO SCENARIO 2: PRIVILEGED USER DATA BREACH

Hackers need privileged access to carry out their malicious activities e.g. installing malicious software or even disabling hardware and/or vital software. In fact, there can be so many shared privileged accounts within a large enterprise that many organisations don't even know where all those accounts reside or who has access to them. Having an efficient and adaptive privilege management system in order to automatically identify these accounts and then bring them under management system and audit access to them will prevent privileged user account misuse. In Demo Scenario 2, in order to identify privileged user data breach, we look at the threats involved in taking on temporary staff over a busy period and the access these staff would need to acquire confidential files. This is done by going through the following steps: 1) identify privileged user accounts within the organisation, 2) identify sensitive files that could only be accessed by those privileged user accounts, and 3) monitor the access to those sensitive files by temporary staff. For example, in Demo Scenario 2, Timmy who is a temporary member of staff with privileged access, accesses a file and folder that he does not need access to.

3.3 Demo Scenario 3: Endpoint Security Processing

Endpoint security management is a policy-based network security that requires endpoint devices to comply with the security policy within an organisation. In Demo Scenario 3, we focus on security processing of company's desktop computers. This means monitoring suspicious activities by the company's staff such as: turning off/disabling system's antimalware e.g. popup blockers, anti-spyware, anti-spam, host-based firewalls and anti-viruses. For example, in Demo Scenario 3, Laura, who is a member of the company's sales team, deactivates the anti-virus software on her computer.

3.4 DEMO SCENARIO 4: SHADOW IT RISK

Shadow IT refers to Information Technology (IT) projects that are purchased, downloaded, installed, used or managed outside or without the knowledge of an organisation's IT department. Shadow IT has grown exponentially in recent years partly due to the good quality of applications in the Cloud and partly due to the rapid rise in Software as a Service (SaaS) development such as: Dropbox, Cisco WebEx, Google Apps, Salesforce, Skype, and Microsoft Office 365. A given SaaS may or may not offer strong security protections e.g. identity management or data encryption. Therefore it can expose an organisation to data loss risk and all sorts of security-related threats. In Demo Scenario 4, we focus on Shadow IT risk imposed by employees within an organisation due to using unknown/unauthorised SaaS. For example, in Demo Scenario 4, Rebecca, who is a member of the company's engineering team, installs Dropbox and Skype in order to perform unauthorised back-ups and unapproved uploads to the Cloud, respectively.

3.5 Demo Scenario 5: Data Security

Data Security refers to protective measures applied to prevent unauthorised access to resources. It also protects data from corruption. Backups, data encryption, authentication and data masking are the commonly encountered data security techniques. In Demo Scenario 5, we focus on monitoring resources that a third party staff is not supposed to access or make a copy from. For example, in Demo Scenario 5, Colin, a third party contractor, accesses and copies data that he is not supposed to.

3.6 Demo Scenario 6: Protect Sensitive Folders

While the public information can be available to anyone, the sensitive information can only be released to those who have a legitimate need to know and can be further divided into: payment card information, personally identifiable information and health information. For example, personally identifiable information is any information about any individual and/or linked/linkable to any individual that is maintained by an organisation and can be used to trace an individual's identity e.g. place of birth. In Demo Scenario 6, we focus on protecting folders with sensitive information. For example, in Demo Scenario 6, Colin, a third party contractor, accesses one of the staff's medical records which he should have no need to access to.

4. ANALYSING THE DATA

Given the six demo scenarios that we defined in the previous section, the original dataset has been given by ZoneFox and includes Charlotte, Rebecca, Laura, Timmy and Colin's user profiles captured in four consecutive days. The original dataset is in .CSV format and contains 2643 lines of raw data including eight features: Date-Time, machine_ID, user_ID, application_ID, application_type, action_ID, action_type and resource_location. Each line of the dataset identifies an action done by one of the users from a specific computer on a particular date-time. The raw data has been gone through the following phases.

4.1 Data Pre-processing

Raw data is often incomplete, inconsistence, noisy and/or lacking certain behaviors/styles and is likely to contain and/or generate error if fed intact into data mining and/or machine learning algorithms. Data pre-processing is a proven method of resolving such issues and includes various steps such as: data cleaning, parsing, correcting, standardizing, matching, consolidating and data staging. In this paper, it includes the following steps.

4.1.1 Outlier Identification

In this paper, outlier refers to insider threats or more specifically it refers to employees who preform digital tasks which they are not supposed to do and/or they should have no need to do. Our focus in outlier identification step is to detect suspicious user's activity based on six demo scenarios defined in the previous section and original .CSV file that includes 2643 lines of user's activity within the organisation. To achieve this, we investigated Laura, Timmy and Colin from three groups of: permanent staff, temporary staff and third party staff, all respectively. The process of outlier identification is done manually by: going through each event in the original .CSV file, comparing it with six demo scenarios and then marking it as either 1, representing an outlier event, or 0, representing a non-outlier event. As it is depicted in Table 1, the outlier identification phase results in total of 33 outliers out of 2643 events. Hence, the outlier identification step adds one more feature of "Outlier" with Boolean value of either 0, representing non-outlier, or 1, representing outlier, to the original .CSV file which results in nine features in total. In this paper labelling the entire dataset through outlier identification phase is done for two main reasons. Firstly, this will provide a platform to evaluate the unsupervised approach results e.g. by comparing them with the supervised approach results. Secondly, it will assist us to implement the semi-supervised approach that has a potential to improve the accuracy rate for insider threat detection.

4.1.2 Data Conversion

Data mining and/or machine learning tools were unlikely to process any data correctly and/or produce any accurate results if the data does not follow a similar data type. Therefore, we employed a data conversion step on our original dataset in order to transform them into a similar data type. The data conversion step has been engaged on each eight features and transforms them into the numeric value. For example, in the main dataset, the Date-Time filed for each event contains standard date and time format including both numeric and text character. For instance, 2016-02-23T16:26:33Z indicates a user event that has happened on 23th of February 2016 at 16:26:33 hours. This includes two characters: T, which can be read as an abbreviation for Time, and Z, which stands for zero-time zone as it is offset by 0 from the Coordinated Universal Time (UTC). For Date-Time data conversion, we first split it to date and time which gives us: 2016-02-23T and 16:26:33Z for our running example above. We then removed T and Z characters which gives us: 2016-02-23 and 16:26:33. Then, we formatted the date to: 23/02/2016 while time stays the same: 16:26:33. We then converted date and time to a UNIX timestamp which results in 1456185600 for the date and 59193 for the time. In the last step, we combined them together which gives us: 1456244793. Furthermore, in our original dataset, each machine_ID is a combination of numbers, uppercase and lowercase letters e.g. 4RcZBZz. For machine_ID data conversion, we defined 15,000 – 19,000 range from which an integer value has been assigned to each user computer. Likewise, 1000 – 1500 range has been identified for the user_IDs in the dataset. Similarly, 20 – 99 range has been assigned to application_ID, 200 - 499 to action_ID and 0-4 to resource_location. Moreover, in terms of application type and action type, 0 represents systems application and process related actions and 1 represents user application and file related actions, both respectively.

4.1.3 Normalisation

Normalisation gives the data a fair chance of comparison without creating a bias in the results. This means it removes the possibility of implying that one feature with a higher value is more important than the other feature with a lower value. One of the popular normalisation formula is depicted in Equation 1 where x is a given variable to be normalised, x_{min} is the minimum value, x_{max} is the maximum value and x_{new} is the new value for the variable after normalisation. In this paper, we used the same formula for the normalisation phase which scales the data from the range of: [0, 19000] into the range of [0,1]. This phase applied to all the eight features in our original .CSV file.

5. Implementations and Results

In this paper, we use an unsupervised competitive machine learning algorithm with a particular focus on SOM model which is a neural network-based clustering technique that transforms a continuous high dimensional input space mapped to a discrete low dimensional output space. In unsupervised machine learning algorithms, the networks learn to form their own classifications of the input data without any external help. In a competitive machine learning algorithm such as SOM, the output neurons compete amongst themselves to be activated given that only one neuron can be activated at a time. However, before using SOM, we employ PCA on our entire dataset. PCA, as the name says, finds the principal components of data rather than for example its normal x-y axis. The principal components of data are the directions where there is the most variance through which the date is most spread out. Therefore, we first briefly discuss PCA and SOM follows by the SOM results with and also without employing PCA.

5.1 Principal Component Analysis (PCA)

PCA is a method of extracting few important variables from a large set of variables in a given dataset usually with three or higher dimensions. Therefore, the general idea is to construct some principal components which satisfactorily explains most of the variability in the data. In general, PCA goes through five steps as follows. The first step is to calculate the mean for each dimension e.g. for a dataset with two dimensions of x and y, this step gives us two values of x_{mean} and y_{mean} each representing the mean value for the associated dimension. The second step is to subtract the mean value from each individual sample in each dimension e.g. for our running example, this step gives us $[x_1 - x_{mean}, \ldots x_n - x_{mean}]$ and $[y_1 - y_{mean}, \ldots, y_n - y_{mean}]$.

Table 1. Outlier Identification

Employee Name	Total Events 441/2643	Outlier Events 33/2643
Charlotte	146	6
Rebecca	75	5
Colin	57	11
Laura	135	1
Timmy	28	10

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Equation 1. Normalisation

The third step is to calculate the covariance matrix which is represented by an n x n matrix where n represented the number of the dimensions. To calculate covariance matrix, we need to: 1) calculate the variance for each dimension individually which is a measure of the spread of the data and 2) covariance which is a measure of how much each two dimension varies from the mean with respect to each other. In our running example, this gives us variance(x) and variance(y) along with covariance (x,y). Then, the covariance matrix is then represented by:

$$Covariance\ matrix = \begin{bmatrix} variance(x) & covariance\ (x,y) \\ covariance\ (x,y) & variance(y) \end{bmatrix}$$

Equation 2. Covariance Matrix

The fourth step is to calculate the eigen values and eigen vectors on covariance matrix. To explain eigen value and eigen vector, imagine:

$$A \cdot v = \lambda \cdot v$$

Equation 3. Eigen value and eigen vector

Where A is an n x n matrix, v is a n x 1 vector and λ is a scalar (constant). Any value of λ for which this equation has a solution is known as an eigenvalue and v is known as a eigen vector. For PCA calculations, A represents the covariance matrix and n represents the number of dimensions. Given that in our example we have two dimensions (x, y), there will be two eigen vectors of v_1 and v_2 and two eigen values of λ_1 and λ_2. The last step is the actual transformation using some of the eigen vectors of the covariance matrix. In the next section, we explain the result from PCA on our dataset.

5.1.1 Results from PCA

In this paper, we employed PCA on our insider threat dataset which is a matrix of 2643 x 8 given that we have 2643 observations for 8 variables. This gives us 8 PCs each for a single variable.

In Figure 1, we plotted the Explained returned by the PCA as a column vector. The Explained diagram is an 8 x 1 matrix where each component represents the percentages of the total variance explained by each PC. Given that we have eight variables and therefore eight PCs, this means that the first to the eight component in 8 x 1 matrix represents the percentages of the total variance explained by 1st PC to the 8th PC, all respectively. Figure 1 reveals that the 1st PC explained 64.08% of the total variance follows by 17.34%, 11.24%, 4.12%, 2.04%, 0.51%, 0.4% and 0.28% for the 2nd PC to 8th PC, all respectively. Therefore, we can conclude that the first three PCs together explained 92.66% of the total variance in our dataset.

In Figure 2, we plotted the Latent which is the principal component variances for our dataset. Additionally, the Latent is the eigenvalues of the covariance matrix X returned by the PCA as a column vector. This means the Latent is the λ in Equation 3. Given that we have eight dimensions, the eigenvalues of the covariance matrix of 8 x 8 is an 8 x 1 matrix where each component represents a principal component variance for one PC. Figure 2 reveals that the principal component variance for the 1st PC is 0.24 follows by 0.06%, 0.04%, 0.02%, 0.01%, ~0%, ~0%, and ~0% for the 2nd PC to 8th PC, all respectively. Therefore, the first three PCs have the highest variances in our dataset.

In Figure 3, we plotted the Principal Component Coefficients which is also known as Loadings for three PCs: 1st, 4th and 8th. For our, the PCA coefficients is an 8-by-8 matrix in which each column represents coefficients for one PC. The columns are in the order of descending components variance also known as the Latent. This means that the first column, represents the 1st PC, has the highest variance while the last column, represents the 8th PC, has the lowest variance. Additionally, they are the principal component vectors which are the eigenvectors of the covariance matrix. By multiplying the original data by the principal component vectors we will get the projections of the original data on the principal component vector space. We only plot three principal component coefficients here.

Addressing Figure 1, we decided to figure out which variables with which percentages are contributing to each PC. This helps us in our future analysis for example when we use clustering we can only keep variables that contributes the most to the first few/all PCs. For instance, if only three variables are contributing the most to the first three PCs, we can have our observations in three dimensions instead of eight. We calculated this per PC by taking into account the coefficient and latent matrixes. Table 2 reveals the explain variance in an original variable by specific PC.

5.2 Self-Organising Map (SOM)

In a competitive machine learning algorithm, a competition can be implemented by having lateral inhibition connections between the neurons, where neighboring neurons respond less if they are activated at the same time than if one is activated alone. Lateral inhibition will force the neurons to organise themselves, therefore such a network is called a SOM and the activated neuron is called the winning neuron. In general, the SOM algorithm includes four components: Initialisation, Competition, Cooperation and Adaptation. Suppose that we have a continuous high dimensional input space (e.g. 2 Dimensional) and want to map it to a discrete low dimensional output space (e.g. 1D) using SOM. We first need to assign arbitrarily weights to 1D space. After that we need to pick up one data point from 2D space also in a random manner. We then declare the closest neuron in terms of weight from the output as the winning neuron. This makes the winning neuron move towards the data point by a certain amount. The neighboring nodes also move toward the data point however it is not as much as the winning neuron itself. The process continues until the whole continuous high dimensional input space is mapped to a discrete low dimensional output space.

5.2.1 Results from SOM

In this paper, we use SOM for clustering malicious and benign events from our dataset. The malicious events are 33 in total and it is very small in comparison with the total of 2643 events. For our SOM experiments, we defined three sets of tests in order to identify whether the results from PCA experiments could help us obtain a better output from SOM clustering as follows. In test1, we use all the variables from the dataset without considering the PCA results from the previous sections. This means without taking into account which variable has the highest contribution in percentages for each PC. However, in test2, we consider the six variables that have the highest influence on the first three PCs: 1st PC, 2nd PC and 3rd PC. This means the first two variables with highest percentages for each PC, Table 2. This is followed by the last test, test3, where we have taken into account only the first three variables. This means variable7 for the 1st PC, variable5 for the 2nd PC and finally variable1 for the 3rd PC, Table 2. The results from all three sets of experiments are depicted in Figure 4 which shows the number of benign events which are wrongly clustered with malicious events. For example, the first three bars indicate that the total number of benign data points that have been clustered as malicious is 58 for all three tests. To understand Figure 4 better, we then calculated the total clustering accuracy for the malicious events for three sets of experiments. This is: 16.31% for test1, 14.50% for test2 and 19.00% for test3. Therefore, test3 was successful in achieving a slightly better accuracy when it is compared with test1 and test2. This means when we use only three variables, i.e. variable7 from the 1st PC, variable5 from the 2nd PC and variable1 from the 3rd PC, we get slightly better clustering accuracy.

6. Conclusion and Future Work

In this paper, we use SOM which is an unsupervised competitive machine learning algorithm in conjunction with PCA for insider threat detection within an organisation on a very unbalanced dataset. The reason for using this combination is to show the impact of PCA on SOM clustering accuracy. For this, we first run PCA on our dataset on which the entire data has been standardised first and normalised second. We then set up three tests to evaluate the PCA results on SOM clustering accuracy. Addressing the captured results, the PCA improves the clustering accuracy of SOM by nearly 4% when trained only with three variables which have the highest percentages in terms of contribution to each PC. Our future work is to employ this approach on a real-time and a balanced dataset.

Table 2. Variance in each variable explained by each PC

	1st PC	2nd PC	3rd PC	4th PC	5th PC	6th PC	7th PC	8th PC
Var1	0.194	0.045	0.554	0.204	0.000	0.000	0.000	0.000
Var2	0.585	0.007	0.011	0.012	0.381	0.001	0.000	0.000
Var3	0.744	0.007	0.154	0.086	0.006	0.000	0.000	0.000
Var4	0.509	0.411	0.052	0.000	0.000	0.024	0.001	0.001
Var5	0.410	0.468	0.095	0.011	0.001	0.011	0.001	0.000
Var6	0.770	0.137	0.040	0.017	0.000	0.005	0.007	0.021
Var7	0.841	0.126	0.024	0.004	0.000	0.000	0.000	0.002
Var8	0.723	0.207	0.004	0.003	0.000	0.003	0.053	0.004

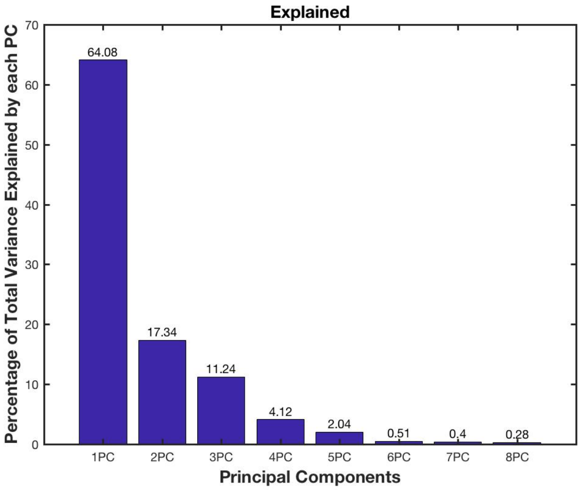

Figure 1. Explained

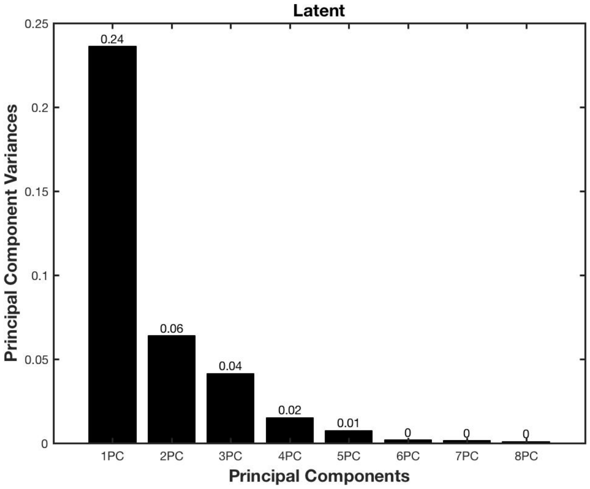

Figure 2. Latent

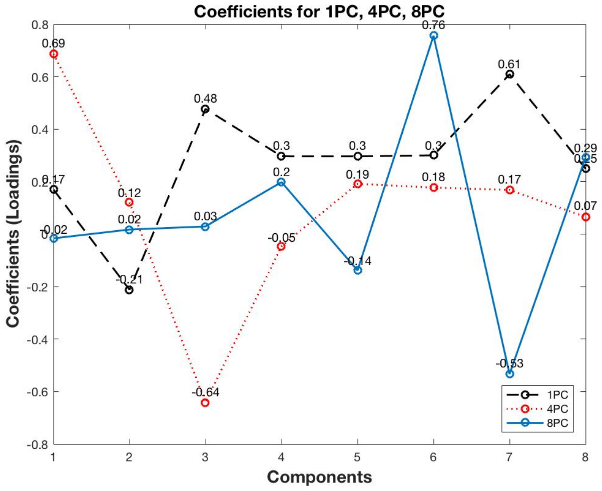

Figure 3. Principal component coefficients

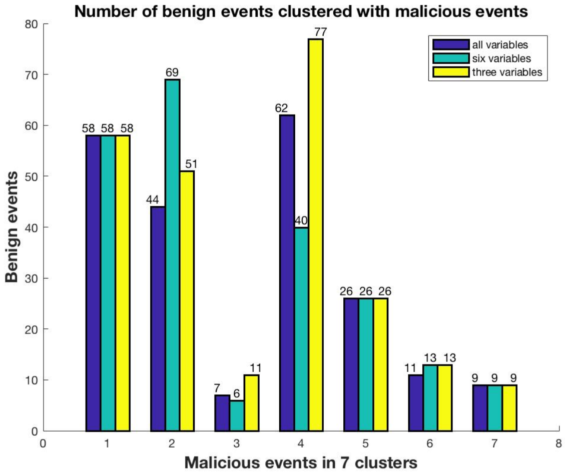

Figure 4. Number of benign events clustered with malicious events

REFERENCES

[1] Singh, A., & Patel, S. S. Applying Modified K-Nearest Neighbor to Detect Insider Threat in Collaborative Information Systems.

[2] Parveen, P., Evans, J., Thuraisingham, B., Hamlen, K. W., & Khan, L. (2011, October). Insider threat detection using stream mining and graph mining. In Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on (pp. 1102-1110). IEEE.

[3] Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2017). Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams.

[4] Böse, B., Avasarala, B., Tirthapura, S., Chung, Y. Y., & Steiner, D. (2017). Detecting Insider Threats Using RADISH: A System for Real-Time Anomaly Detection in Heterogeneous Data Streams. IEEE Sytems Journal.

[5] Hashem, Y., Takabi, H., GhasemiGol, M., & Dantu, R. (2016). Inside the Mind of the Insider: Towards Insider Threat Detection Using Psychophysiological Signals. Journal of Internet Services and Information Security (JISIS), 6(1), 20-36.

[6] Zonefox; availabe on: https://zonefox.com/; last accesed: 1 September 2017

[7] Legg, P. A., Buckley, O., Goldsmith, M., & Creese, S. (2015). Automated insider threat detection system using user and role-based profile assessment. IEEE Systems Journal.

[8] Security, M. I. (2015). Grand Theft Data. Retrieved from Mcafee : http://www.mcafee.com/us/resources/reports/rp-data-exfiltration.pdf.

[9] Gavai, G., Sricharan, K., Gunning, D., Hanley, J., Singhal, M., & Rolleston, R. (2015). Supervised and unsupervised methods to detect insider threat from enterprise social and online activity data. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 6(4), 47-63.

[10] Alexander, H. (2017, 5 17). Who is Chelsea Manning and why is she being released from prison? Retrieved from Telegraph: http://www.telegraph.co.uk/news/2017/05/17/chelsea-manning-released-prison/

[11] Cappelli, D. M., Moore, A. P., & Trzeciak, R. F. (2012). The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud). Addison-Wesley Professional.

[12] SCMP. (2017, June 8). China arrests 22 over sale of Apple private data. Retrieved from South China Morning Post: http://www.scmp.com/news/china/society/article/2097487/chinese-apple-staff-suspected-selling-personal-data.

[13] Haystax Technology. (2017). Insider Attacks: Industrial Survey. Retrieved from Haystax Technology: https://haystax.com/blog/ebook/insider-attacks-industry-survey