Enhanced-PCA based Dimensionality Reduction and Feature Selection for Real-Time Network Threat Detection

Pournima More
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram, Andhra Pradesh, India
pournima.more1@gmail.com

Pragnyaban Mishra
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram, Andhra Pradesh, India
pragnyaban@kluniversity.in

Abstract-**With the rise of the data amount being collected and exchanged over networks, the threat of cyber-attacks has also increased significantly. Timely and accurate detection of any intrusion activity in networks has become a crucial task. While manual moderation and programmed logic have been used for this purpose, the use of machine learning algorithms for superior pattern mapping is desired. The system logs in a network tend to include many parameters, and not all of them provide indications of an impending network threat. The selection of the right features is thus important for achieving better results. There is a need for accurate mapping of high dimension features to low dimension intermediate representations while retaining crucial information. In this paper, an approach for feature reduction and selection when working on network threat detection is proposed. This approach modifies the traditional Principal Component Analysis (PCA) algorithm by working on its shortcomings and by minimizing false detection rates. Specifically, work has been done upon the calculation of symmetric uncertainty and subsequent sorting of features. The performance of the proposed approach is evaluated on four standard-sized datasets that are collected using the Microsoft SYSMON real-time log collection tool. The proposed method is found to be better than the standard PCA and FAST methods for data reduction. The proposed approach makes a strong case as a dimensionality reduction and feature selection technique for reducing time complexity and minimizing false detection rates when operating on real-time data.**

Keywords-principal component analysis; fast clustering; dimensionality reduction; machine learning; network security

I. INTRODUCTION AND RELATED WORK

Network security is of utmost importance, especially for companies or foundations. Any security breach will be characterized by a pattern in the network logs preceding the attack and these patterns, if detected accurately, can help diverting major mishaps [1]. Previous approaches have relied upon the use of Security Intelligence and Management Systems (SIEMs) coupled with manual moderation for scanning network threats. SIEMs are associated with Security Operations Center (SOC) to whom they report these threats [2-3]. They conform to the laws on risks and regulation criteria and make use of predefined rules for catching any network breach or Incident Response (IR). However, similar to manual moderation, there are certain limitations in the use of SIEMs. They operate on a static set of vulnerability rules and hence are not able to detect any trends of a novel attack. Further, they are associated with an operational overhead and hence come up short when working with real-time data. The performance of SOCs has worsened over the years and they are no longer sufficient when working with large scale networks. Machine learning algorithms help in deriving rules and making accurate predictions even on previously unseen data. The behavioral features of the system logs can be used for attack detection. However, the system logs consist of many parameters and not all of them contribute to the detection of network threats. Previously, methods like Principal Component Analysis (PCA) and FAST clustering have been used for eliminating the redundant features from high dimensional data [3, 4]. While PCA focuses on the derivation of a set of orthogonal eigen vectors, FAST clustering focuses on an efficient but less effective grouping of features. In this paper, an approach for dimensionality reduction and feature selection is proposed which ameliorates the shortcomings of the PCA algorithm and improves the performance of the system. Specifically, FAST and PCA approaches were combined in a novel way and subsequently machine learning algorithms were used for detecting anomalies that deviate from normal configurations, hence predicting possible threats to the system. The main contributions in this paper are:

- The formulation of a new approach for dimensionality reduction and feature selection that improves standard conventional approaches.
- The combination of FAST and PCA algorithms in a unique manner for efficient, yet effective dimension mapping.
- A network threat detection approach that has better performance than the previous approaches when working on a real-time data set.

There has been ample work in the past on anomaly detection on unseen data [5-9]. Most of this work can be segregated into three types based on their nature: statistical,

Corresponding author: Pournima More

distance, and density-based techniques. Each of these three methods has some shortcomings. Statistical methods often assume univariate distribution which is most often not true with real-world data. Distance-based methods struggle with overlapping data, while density-based methods may be useful but are computationally very expensive. Authors in [10] proposed a primitive regression method to detect the presence of unauthorized users masquerading as registered users by comparing their activity to the previous actions from those accounts. Various unsupervised learning algorithms have been applied, however, most of them have huge memory requirements [2, 11]. Clustering has been used [12-14], but incorrect grouping would lead to higher risk of false negatives. As a result, dimensionality reduction becomes a crucial part of the process and thus the majority of approaches have focused on the use of PCA for this task [3, 15]. PCA aims to derive new variables, called Main Components (PCs) as linear input variable combinations so that a few of the new variables reflect the overall variation between the input variables. Authors in [19, 20] provide a network intrusion prevention approach that tackles the problem of controlling high computer network traffic and the time pressure to handle security threats. They use the techniques of multivariate analytics, including clustering and PCA to identify classes in the observed data.

The use of PCA is appealing due to its statistical consistency, faster inference, and effective computation [11]. Yet there has been valid criticism in its use for network intrusion detection due to several reasons like struggle with rare class labels, exponential complexity, etc. [4, 7]. Examples of this are [18, 19] where a combination of recorded variation and the screen-plot process were used for selecting the key components which may be risky as some anomaly amongst lesser-known components may be ignored. The existence of unusual data from regular operations is seen in [20] where the criteria of choosing components for PCA remain unanswered. A large number of variables could be reported to explain network traffic behavior [21]. For example, authors in [22] assume a certain expectation for collection methods for the variable. Typically, every input variable has a non-zero factor on all PCs. However, in practice, it is common that the majority of the component loadings is close to zero [23] and is of less practical significance. Recently there have been proposals of some hybrid approaches [24-26] but they have a higher training and inference cost, with increased complexity. Wormhole attacks have been tackled using separate routing [27] but the method does not work in structured log data. While synthetic oversampling and under sampling approaches may work well on text data [28], they are often detrimental to structured network log data. Hence it can be seen that there is a need for improving the shortcomings of the previous methods and come up with a more accurate and efficient solution.

II. Proposed Methodology

In this paper, we work upon an enhanced feature selection and reduction technique that ameliorates the shortcomings of the previous approaches. The diagrammatic representation of the proposed algorithm is shown in Figure 1. The phases on the algorithm are defined in the following subsections.

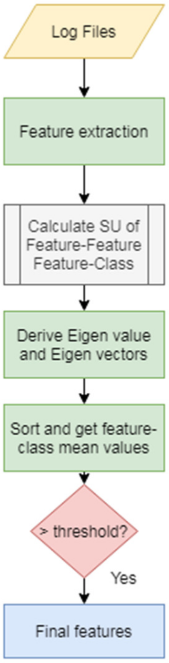
Log Files
Feature extraction
Calculate SU of Feature-Feature Feature-Class
Derive Eigen value and Eigen vectors
Sort and get feature-class mean values
> threshold?
Yes
Final features

Fig. 1. Diagrammatic representation of the proposed algorithm.

A. *Log Files and Feature Extraction*

The Microsoft SYSMON log process collection tool was utilized for extracting the logs from the given system. These logs are collected in real-time and have a certain number of input attributes associated with them. These attributes are described in detail below. The data features are stored together in a matrix format for subsequent processing.

B. *Calculation of Symmetric Uncertainty*

Symmetric Uncertainty (SU) is used to obtain the selection of features by calculating the fitness of the features between the feature and the target class. Symmetric uncertainty is defined as:

$$\mathrm{SU}(X,Y)=\frac{2*\mathrm{Gain}(X\,|\,Y)}{H(X)+H(Y)} \qquad (1)$$

where Gain(X|Y) is the amount by which the entropy of the variable Y decreases. H(X) is the entropy of a discrete random variable X. If the prior probability of each element of X is p(x), then H(X) can be calculated as:

$$\mathrm{H}(X)=-\sum_{x\in X}\mathrm{p}(x)\log_2\mathrm{p}(x) \qquad (2)$$

The entropy value takes care of any associated bias amongst the features with large values and also normalizes them to a range of [0, 1]. An SU value of 1 indicates that the information value of the one variable is fully represented by the other, whereas a 0 value of SU(X, Y) indicates that X and Y are independent variables. Such mathematical representation also helps normalizing continuous features to a discrete form. These SU values are used for the calculation of the eigenvalues and the eigenvectors.

C. Eigenvalues and Eigenvectors

Eigenvalues are used to calculate the dependency of features and their correlation with the class values. The eigenvector with the highest uniqueness is considered as the most characteristic feature implying the highest variance [4]. Similarly, the second most unique vector is considered as the second characteristic principal variable with information retention. In this way, the top *N* eigenvectors are calculated and represented using a co-occurrence matrix. *N* indicates the higher contribution rate amongst all the eigenvectors. If *M* is a $n \times n$ matrix, then *v* is considered as the eigenvector of *M* if:

$$M \times \boldsymbol{v} = \lambda \quad (3)$$

where λ is the eigenvalue associated with *v*. For the given eigenvector *v* of *M*, given a scalar *a:*

$$\mathbf{M} \times a\mathbf{v} = \lambda a\mathbf{v} \quad (4)$$

N different eigenvectors can always be chosen such that they collectively account for a unit length:

$$\sqrt{{v_1}^2 + \ldots + {v_n}^2} = 1 \quad (5)$$

The *n* eigenvectors are always orthogonal to each other. Thus, they can be used as the basis for the formulation of a new *n*-dimensional vector space. It is crucial to analyze the security of information from different unknown sources or attacks [10]. A fixed policy can never detect new different threats that are created. This examination can be done on different network trees. Once the FAST calculations of SU are done, the representation of the method is done by us using the PCA method. A multidimensional hyperspace is hard to visualize. The key objectives of the unsupervised learning approaches are to minimize dimensionality, ranking all identifications based on a composite index, and clustering related identifications based on multivariate attributes together. Since it is difficult to imagine a multidimensional space, PCA is used to scale down the dimensionality of multivariate parameters into three dimensions. In this paper, PCA is used for multivariate analysis. The data set can be represented as a matrix *X* with *n* rows and *m* columns, the sample rows, and the attribute columns. Some multivariate methods presume a structure, while others separate the cases into classes trying to figure out the structure of data. The former is a case of supervised learning while the latter indicates unsupervised learning.

In PCA, the interrelated multivariate parameters are mapped to a set of non-related components, each expressing a distinct linear combination of the main variables. The non-related components extracted are the PCs and are predicted from the covariance matrix's main variables' ownership. PCA aims to obtain providence and minimum dimensionality by extracting the smallest number of PCs that account for most of the variation in the main multivariate information and summarizing the data with minimum information loss. The variance of an attribute is defined as:

$$\mathrm{var}(A_1) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{(n-1)} \quad (6)$$

In this method, the last principal component score needs to be zero. All the given variables are scattered on a hyper plane. If there is any update in the interrelations, there may be an extension of the information outside the hyper plane. This is reflected in the updates of the principal component scores that were previously zero. Nil scores are the most sensitive to updates in the interrelations. The main component scores are powerful in observing any updates to the information. PCA performs normal data circulation by selecting a suitable orthogonal coordinate. The range of the main rectangular factor scores is more suitable for expressing normal data distribution than those of the sensed and actuated initial orthogonal coordinates.

D. Threshold Calculations

After the eigenvectors are calculated, they are sorted in descending order and the feature class mean value is calculated. This value is used for deciding the threshold value for obtaining the features. The following algorithm indicates the entire process followed by the enhanced PCA algorithm.

```
Input: Input attribute matrix $X \in x_1, x_2, \ldots, x_n$, output $Y$, threshold $t$
Output: Reduced representation $S$
For each $x_i$ in $X$ do
    Calculate H($x_i$)
    Calculate SU($x_i$,$Y$)
    Calculate Variance($x_i$)
Endfor
Sort all variances in descending order
Calculate eigenvalues $\lambda_1, \lambda_2, \ldots. \lambda_n$
Calculate subsequent $V \in v_1; v_2; :::; v_n$
For each $v_i$ in $V$ do
    If $v_i > t$ do
        $S$.append($v_i$)
    Endif
Endfor
return $S$
```

III. Dataset Description

For evaluation using a real-time dataset, data were collected from logged data using the Microsoft Sysmon data collector tool [30]. It provides the information of certain parameters present in the network, considered as the input variables for the network intrusion detection task. For a more generalized evaluation and elimination of bias, we collect four different sets of data each of varying sizes. The individual sizes of each of the dataset are:

- Dataset 1: 1000
- Dataset 2: 1200
- Dataset 3: 800
- Dataset 4: 361

The total dataset size accounts for 3361 samples. Each of the datasets is split into a training set and a testing set in a 3:2 ratio. For every sample in the dataset, there are 22 variables associated with it indicating the corresponding configuration. These variables are described in Table I.

TABLE I. DESCRIPTION OF THE INPUT ATTRIBUTES

Parameter	Description
Process name	1. Name of the process
Login time	Login time with 8 segmentations, three hours each: 2. 12pm-3pm 3. 3pm-6pm 4. 6pm-9pm 9. 9am-12pm
Browsing history	10. URLs being browsed (if any)
Network transfer	11. Uploading 12. Downloading
Days of the week	13. Current day
Frequency	14. No. of executed processes
Spam mail keywords	15. Credit card 16. Loan 17. Offer 18. Monsoon 19. Sale 20. Winning 21. Money 22.Prize

IV. RESULTS AND DISCUSSION

The obtained results are presented on two different levels: the output of the dimensionality reduction and the obtained performance on task-specific performance metrics. While the output variables received by the dimensionality reduction algorithm can be variable, we judge their effectiveness by comparing the results with those obtained by the models that use other dimensionality reduction methods. Table II indicates the output variables determined by the proposed approach to be deemed most important and relevant to the task in hand. The threshold was set to 66% and thus the total number of variables was reduced to 15.

TABLE II. OUTPUT PARAMETERS DERIVED BY THE PROPOSED ALGORITHM

Parameter No.	Output parameter name
1	Name of the process
9	9am-12pm
3	3pm-6pm
11	Uploading
7	3am-6am
15	Credit card
5	9pm-12pm
17	Offer
2	12pm-3pm
10	Browsing history
6	12am-3am
12	Downloading
21	Money
19	Sale
16	Loan

The performance of the proposed approach is evaluated with two different metrics: Accuracy and Inference time. These are defined in (7) and (8):

$$\text{Accuracy}=\frac{\text{No: of correctly predicted samples}}{\text{Total no: of samples}} \quad (7)$$

$$\text{Inference time} = T_{Output} - T_{Input} \quad (8)$$

Inference or prediction time is defined as the amount of time required for the system to output the prediction for a given input set. It is found by subtracting the time frame when the input was given from the time frame when the output was predicted. The results of the proposed approach for each of the four datasets for the three methods (FAST, PCA and enhanced PCA) are compared. The accuracy obtained for the three approaches is shown in Table III. It can be seen that the proposed approach has outperformed the standard approaches and has a healthy growth over the use of the normal PCA approach. In Table IV the evaluation of the tested methods in terms of Inference time is shown. It can be seen that while the proposed approach is not the best regarding inference time, it still performs better than the PCA algorithm. While the FAST algorithm has faster inference time, the proposed approach is better in terms of accuracy, which is more important in the case of network intrusion detection systems. The graphical comparison of the three approaches in terms of accuracy and inference time is shown in Figures 2-3. Thus our proposed approach has been able to get the right mix of effectiveness and efficiency.

TABLE III. ACCURACY (%) OF THE TESTED ALGORITHMS

Dataset	FAST	PCA	Enhanced PCA (proposed)
Dataset 1	90	87	92
Dataset 2	90	88	92
Dataset 3	88	88	89
Dataset 4	94	90	96

TABLE IV. INFERENCE TIME (ms)

Dataset	FAST	PCA	Enhanced PCA (proposed)
Dataset 1	350	420	370
Dataset 2	360	530	380
Dataset 3	350	440	430
Dataset 4	270	315	260

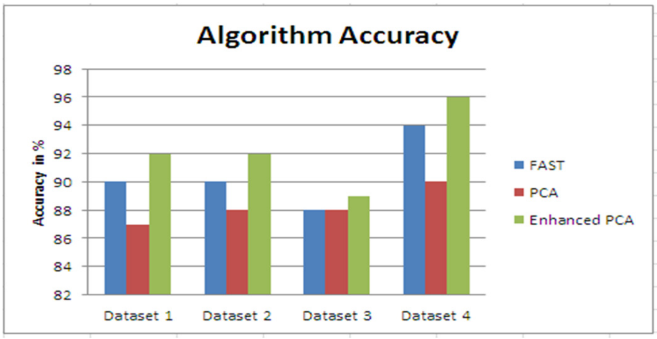

Fig. 2. Accuracy comparison.

V. CONCLUSION

In this paper, a new approach was presented for the dimensionality reduction and the subsequent prediction of any intrusion threats to a network system. We were able to minimize time complexity and false detection rates, especially on a real-time data set.

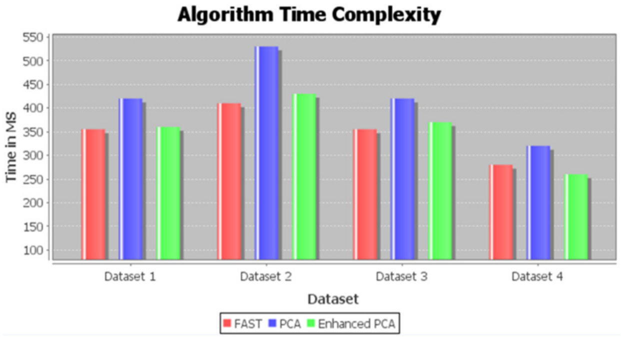

Fig. 3. Inference time comparison.

The proposed approach derived inspiration from the traditional PCA algorithm but ameliorated its shortcomings and improved further its performance. Specifically, we made use of symmetric uncertainty, entropy, and associated factors to boost the working of the feature selection PCA algorithm. The Sysmon tool was used for the collection of data. The performance of the proposed approach was evaluated on four different datasets and was compared with the standard PCA and FAST approaches. The proposed approach was found to be more accurate while possessing a satisfactory inference time. An increment of over 2% was observed in terms of accuracy as compared to the standard approach. The reduction time was faster than the regular PCA approach by more than 15% in all cases. The proposed approach was proved to be a more preferred method than the previous approaches. Future work in this domain includes working upon other machine learning algorithms to couple with the dimensionality reduction method for enhanced results. Other preprocessing methods can also be used so that the initial variables are made more model-friendly. Genetic algorithms can be thought of as an alternative solution for enhanced optimization strategy. The proposed approach can be considered as a small contribution in the creation of timely and accurate network intrusion detection systems.

REFERENCES

[1] S. Staniford-Chen and L. T. Heberlein, "Holding intruders accountable on the Internet," in *IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 1995, pp. 39–49, doi: 10.1109/SECPRI.1995.398921.

[2] S.-J. Horng *et al.*, "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Systems with Applications*, vol. 38, no. 1, pp. 306–313, Jan. 2011, doi: 10.1016/j.eswa.2010.06.066.

[3] M. L. Shyu, S. C. Chen, K. Sarinnapakorn, and L. . W. Chang, "A Novel Anomaly Detection Scheme Based on Principal Component Classifier," 2003, pp. 172–179.

[4] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," in *ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, New York, NY, USA, Jun. 2007, pp. 109–120, doi: 10.1145/1254882.1254895.

[5] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: 10.1145/1541880.1541882.

[6] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, Aug. 2008, pp. 444–452, doi: 10.1145/1401890.1401946.

[7] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Conditional Anomaly Detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 5, pp. 631–645, May 2007, doi: 10.1109/TKDE.2007.1009.

[8] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, May 2000, pp. 93–104, doi: 10.1145/342009.335388.

[9] A. T. Siahmarzkooh, S. Tabarsa, Z. H. Nasab, and F. Sedighi, "An Optimized Genetic Algorithm with Classification Approach used for Intrusion Detection," 2015. /paper/An-Optimized-Genetic-Algorithm-with-Classification-Siahmarzkooh-Tabarsa/b0e239298e7c6d8aa0e813a12fe55a2d12673e29 (accessed Sep. 12, 2020).

[10] W. Dumouchel and M. Schonlau, "A Comparison of Test Statistics for Computer Intrusion Detection Based on Principal Components Regression of Transition Probabilities," in *Proceedings of the 30th Symposium on the Interface: Computing Science and Statistics*, 1998, pp. 404–413.

[11] Z. Muda, W. Yassin, M. N. Sulaiman, and N. I. Udzir, "Intrusion detection based on K-Means clustering and Naïve Bayes classification," in *7th International Conference on Information Technology in Asia*, Kuching, Sarawak, Malaysia, Jul. 2011, pp. 1–6, doi: 10.1109/CITA.2011.5999520.

[12] A. T. Siahmarzkooh, J. Karimpour, and S. Lotfi, "A Cluster-based Approach Towards Detecting and Modeling Network Dictionary Attacks," *Engineering, Technology & Applied Science Research*, vol. 6, no. 6, pp. 1227–1234, Dec. 2016.

[13] J. Karimpour, S. Lotfi, and A. T. Siahmarzkooh, "Intrusion detection in network flows based on an optimized clustering criterion," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 25, no. 3, pp. 1963–1975, May 2017.

[14] A. T. Siahmarzkooh, In press. A GWO-based Attack Detection System Using K-means Clustering Algorithm (No. TRKU-11-08-2020-10987), Technology Reports of Kansai University.

[15] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," in *4th ACM SIGCOMM Conference on Internet Measurement*, New York, NY, USA, Oct. 2004, pp. 201–206, doi: 10.1145/1028788.1028813.

[16] C. Taylor and J. Alves-Foss, "NATE: Network Analysis of Anomalous Traffic Events, a low-cost approach," in *Proceedings of the 2001 workshop on New security paradigms*, New York, NY, USA, Sep. 2001, pp. 89–96, doi: 10.1145/508171.508186.

[17] C. Taylor and J. Alves-Foss, "An empirical analysis of NATE: Network Analysis of Anomalous Traffic Events," in *Proceedings of the 2002 workshop on New security paradigms*, New York, NY, USA, Sep. 2002, pp. 18–26, doi: 10.1145/844102.844106.

[18] W. Wang and R. Battiti, "Identifying intrusions in computer networks with principal component analysis," in *First International Conference on Availability, Reliability and Security*, Vienna, Austria, Apr. 2006, pp. 1–8, doi: 10.1109/ARES.2006.73.

[19] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "When randomness improves the anomaly detection performance," in *3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Rome, Italy, Nov. 2010, pp. 1–5, doi: 10.1109/ISABEL.2010.5702782.

[20] R. Kwitt and U. Hofmann, "Unsupervised Anomaly Detection in Network Traffic by Means of Robust PCA," in *International Multi-Conference on Computing in the Global Information Technology*, Guadeloupe City, Guadeloupe, Mar. 2007, pp. 37–37, doi: 10.1109/ICCGI.2007.62.

[21] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 227–261, Nov. 2000, doi: 10.1145/382912.382914.

[22] M. Koeman, J. Engel, J. Jansen, and L. Buydens, "Critical comparison of methods for fault diagnosis in metabolomics data," *Scientific Reports*, vol. 9, no. 1, Feb. 2019, doi: 10.1038/s41598-018-37494-7, Art. no. 1123.

[23] H. Zou, T. Hastie, and R. Tibshirani, "Sparse Principal Component Analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, Jun. 2006, doi: 10.1198/106186006X113430.

[24] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection," in *Proceedings of the Australasian Computer Science Week Multiconference*, New York, NY, USA, Jan. 2018, pp. 1–6, doi: 10.1145/3167918.3167951.

[25] A. J. Malik, W. Shahzad, and F. A. Khan, "Network intrusion detection using hybrid binary PSO and random forests algorithm," *Security and Communication Networks*, vol. 8, no. 16, pp. 2646–2660, 2015, doi: 10.1002/sec.508.

[26] Y. Zhong *et al.*, "HELAD: A novel network anomaly detection model based on heterogeneous ensemble learning," *Computer Networks*, vol. 169, Mar. 2020, doi: 10.1016/j.comnet.2019.107049, Art. no. 107049.

[27] F. Rezaei and A. Zahedi, "Dealing with Wormhole Attacks in Wireless Sensor Networks Through Discovering Separate Routes Between Nodes," *Engineering, Technology & Applied Science Research*, vol. 7, no. 4, pp. 1771–1774, Aug. 2017.

[28] P. Ratadiya and R. Moorthy, "Spam filtering on forums: A synthetic oversampling based approach for imbalanced data classification," *arXiv:1909.04826 [cs, stat]*, Sep. 2019, Accessed: Sep. 12, 2020. [Online]. Available: http://arxiv.org/abs/1909.04826.

[29] P. More and M. P. Mishra, "Machine Learning for Cyber Threat Detection," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1.1, pp. 41–46, 2020, doi: 10.30534/ijatcse/2020/0891.12020.

[30] M. Russinovich and T. Garnier, "Sysmon v11.11," Jul. 15, 2020. https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon (accessed Sep. 12, 2020).