ISSN: 0067-3064

Enhancing Cyber Threat Categorization with Artificial Intelligence: A Novel Clustering-Based Classification Framework

Matei Vasile CAPILNAS[1,*], Adriana Mihaela COROIU[1]

[1] Computer Science Department, Cluj-Napoca, Romania, 400084, matei.capilans@ubbcluj.ro; adriana.coroiu@ubbcluj.ro

Abstract
This study investigates the application of artificial intelligence (AI) to improve cyber threat classification using clustering techniques. Leveraging the NF-UNSW-NB15-v2 dataset, the research addresses challenges such as data imbalance and overlapping attack patterns. The methodology integrates dimensionality reduction via Principal Component Analysis (PCA) and clustering using KMeans, focusing on features like transport layer ports, DNS query types, and network throughput. The experiments highlight the clustering algorithm's ability to identify inherent patterns within attack categories, though difficulties persist in distinguishing closely related attack types. Despite the imbalanced dataset, clustering by attack type revealed significant insights, enhancing the nuanced analysis of cyber threats. Evaluation metrics, including the silhouette score, emphasize areas for refinement. The findings demonstrate the potential of AI-driven clustering to complement existing cybersecurity frameworks, offering a pathway for more effective intrusion detection systems. This research underscores the importance of combining clustering with additional techniques to improve classification accuracy, advancing the capability of AI in addressing evolving cybersecurity threats.

Keywords: cybersecurity, clustering, artificial intellicence, KMeans, cyber attack

*Correspondent author: matei.capilnas@ubbcluj.ro

Introduction

We have chosen to adopt a more objective perspective and conduct a thorough examination of the current state of the information technology industry to gain a comprehensive understanding of our present circumstances. New applications and technological advancements are being introduced annually, driven by the rapid acceleration of innovation. While these advancements have made life more convenient and efficient, they have also introduced significant challenges. One of the most serious challenges we face is the rise in online criminal activity. The internet provides an ideal environment for criminals to operate anonymously and execute a wide range of illicit activities. The shift from the physical world to the digital domain has enabled criminals

to exploit digital connectivity to conduct various operations. Hackers now use sophisticated methods to exploit vulnerabilities in digital systems, secretly stealing financial resources and contributing to the surge in malicious financial activity. Furthermore, the dark corners of the internet have become a safe haven for illegal drug trafficking, where transactions occur beyond the reach of law enforcement.

State of the art

The study titled “Intrusion Detection in IoT Networks Using Deep Learning Algorithm” (Susilo and Sari, 2020) employed machine learning techniques such as Random Forest (RF) and deep learning methods like convolutional neural networks (CNN) and multilayer perceptron (MLP) to classify attacks in IoT networks. Increasing the batch size notably accelerated computational speed, with a 1.4–2.6× improvement for MLP and 1.8–2.4× for CNN.

The works “URL-based Phishing Attack Detection by CNN” (Nowak et al., 2019) and “Accurate and Fast URL Phishing Detector” (Wei et al., 2020) achieved 99.98% accuracy in detecting phishing scams using CNNs. They added embedding layers to enhance compatibility with mobile devices by adapting URL representations.

In “Analysis of Naive Bayes Algorithm for Email Spam Filtering” (Rusland et al., 2017), the Naive Bayes classifier effectively identified spam emails by employing a bag-of-words technique and Bayes' theorem. Terms like “Free” and “Viagra” were flagged with high spam probabilities, while common words in non-spam emails were flagged with low probabilities.

“A Survey on the Use of Data Clustering for Intrusion Detection” (Bohara et al., 2020) highlighted K-means as the most frequently used clustering method for intrusion detection systems (IDS). It was combined with techniques like hierarchical clustering, fuzzy methods, and decision trees. Evaluation metrics such as detection rate and false-positive rate determined effectiveness, with high detection and accuracy correlating with high efficacy.

Ahmad et al. (2021) provided a summary of machine learning and deep learning techniques in network intrusion detection systems (NIDS), outlining steps including data preprocessing, model training, and testing. Data imbalance and the complexity of deep learning models posed challenges during training and evaluation phases.

Aung and Min (2018) utilized the K-means and Random Forest algorithms with the KDDCup 99 dataset to identify network attacks. Their findings revealed clustering patterns linked to specific attack types like Denial of Service (DoS) and Probe, while emphasizing the mimicry of normal behavior during intrusions.

Portnoy (2000) introduced a hierarchical clustering algorithm capable of identifying both known and unknown intrusions. While effective, it required manual determination of cluster width (W), which could lead to misclassifications if inaccurately set.

ISSN: 0067-3064

Methodology

Unsupervised learning in computer science poses a significant challenge, especially in clustering or cluster analysis. The primary objective is to group data points based on their similarities without relying on a target variable. This method is particularly valuable for unlabeled datasets where patterns are not immediately apparent, enabling intuitive analysis. Metrics such as Euclidean distance, Cosine similarity, and Manhattan distance are used by clustering algorithms to determine similarity, leading to the formation of homogeneous clusters. This approach uncovers inherent patterns within heterogeneous datasets, offering insights into their underlying distribution (Ikotun et al., 2023).

Machine Learning Techniques

Random Forest

Random Forest (RF) creates multiple independent decision trees and combines their results. Internal nodes use selected features to split datasets into homogeneous subsets, guided by Gini impurity criteria, which identifies features that yield the greatest impurity reduction (Alduailij et al., 2022).

K-Nearest Neighbors (KNN)

KNN classifies data observations based on their proximity to neighboring classes. It is a semi-supervised, non-parametric approach that calculates distances between points to assign labels. The value of K significantly influences the method's accuracy and efficiency (Alduailij et al., 2022).

XGBoost

XGBoost combines regression trees and gradient boosting algorithms. Each successive classifier improves the residuals of the previous one, minimizing complexity and overfitting while ensuring robust predictions (Ben Jabeur et al., 2023).

K-Means Clustering

K-Means is an iterative, unsupervised machine learning technique that partitions data into clusters by associating points with the nearest centroid. The algorithm updates centroids until stability is achieved, optimizing the sum of squared differences within clusters. Techniques like the silhouette score and elbow method help determine the optimal number of clusters (Yuan and Yang, 2019; Habib, 2021).

Computational Experiments and Results

Currently, we inhabit a hyperconnected world in which millions of diverse gadgets incessantly exchange information across many application settings for health, enhancing communication, digital enterprises, and more. Nonetheless, an increase in the number of devices and connections elevates the potential of security vulnerabilities in this context. To mitigate harmful activities and maintain critical security services,

Network Intrusion Detection Systems (NIDSs) serve as the predominant defense mechanism in communication networks (MaganCarrion et al., 2020). The experiments in this paper were done on the NF-UNSW-NB15-v2 data set (Sarhan et al., 2022). The NF-UNSW-NB15 dataset, which adopts a NetFlow-based format, serves as the second iteration of the UNSW-NB15 dataset. This version, labeled NF-UNSW-NB15, has been enhanced by incorporating additional NetFlow features and categorizing entries according to specific attack types. In total, the dataset comprises 2,390,275 data flows, with 95,053 (3.98%) identified as attack instances and 2,295,222 (96.02%) as benign. The attack instances are further subdivided into nine distinct subcategories. The distribution of all flows within the NF-UNSW-NB15-v2 dataset is outlined in the table below (Table 1).

Tabel 1
NF-UNSW-NB15-v2 dataset attacks count

Class	Count
Benign	2295222
Fuzzers	22310
Analysis	2299
Backdoor	2169
DoS	5794
Exploits	31551
Generic	16560
Reconnaissance	12779
Shellcode	1427
Worms	164

We chose to use this dataset due to its complexity and the increasing frequency of its use recently (Tabel 2).

Tabel 2
NF-UNSW-NB15-v2 dataset based articles

Authors	Title	Release Year
Alam, K., Monir, M. F., Hassan, Z., & Habib, M. T. (2024, October).	Optimizing IoT Network Intrusion Detection: A Deep Learning Approach	2024
Bhuiyan, M. H., Alam, K., Shahin, K. I., & Farid, D. M. (2024, September).	A Deep Learning Approach for Network Intrusion Classification	2024
Ba, A., & Adda, M. (2024).	Intrusion Detection in IIoT Using Machine Learning	2024
Yang, C., Wu, L., Xu, J., Ren, Y., Tian, B., & Wei, Z. (2024).	Graph Learning Framework for Data Link Anomaly Detection	2024
Ajagbe, S. A., Awotunde, J. B., & Florez, H. (2023).	Ensuring intrusion detection for iot services through an improved CNN	2023

Engineered benchmark NIDS datasets have been developed due to the challenges associated with acquiring labeled realistic network traffic. A network testbed is constructed to replicate the network behavior of various end nodes. The artificial network envi ronment addresses the security and privacy challenges encountered by realworld networks. Furthermore, categorizing the network flows produced by these controlled settings is more dependable compared to the unpredictable characteristics of realworld networks. Throughout the experiments, both benign network traffic and a range of attack scenarios are produced and executed within the network testbed. During this time, the network packets are collected in their original packet capture (pcap) format and stored on storage devices. Network data features are extracted from the pcap files utilizing suitable tools and methods, resulting in the formation of network data flows. The outcome is a dataset of labeled network flows that illustrates both benign and malicious network behavior (Sarhan et al., 2022). One of the initial challenges encountered during the experimentation phase stemmed from the imbalanced nature of the dataset. With 96% of the data labeled as benign, only a mere 4% represented various cyber attacks. A significant challenge in constructing default prediction models is the problem of imbalanced data. Class imbalance arises when the training samples of one majority class significantly exceed those of the minority class. Studies have shown that algorithms trained on an imbalanced dataset often exhibit prediction bias, leading to subpar performance in the minority class. One

way to handle an imbalanced dataset is to downsample and upweight the majority class. When downsampling, the essential task is to decrease the sample size by selecting a representative subset that captures the overall traits of the entire population. This approach resembles dimension reduction techniques when we concentrate on the features rather than the instances. A widely recognized informed downsampling technique involves utilizing the nearest neighbors. Tomek links (TL) refer to a pair of instances belonging to two distinct classes, characterized by their proximity as determined by a 1-nearest neighbor distance. These pairs can be considered as borderline instances, or one of them may be noise that is prone to misclassification. Consequently, either both samples or one from the majority class is removed, as they do not contribute to the dataset's quality. Additional methods for downsampling that utilize nearest neighbor techniques consist of the Neighborhood Cleaning Rule (NCL) and Wilson's Edited Nearest Neighbor (ENN). ENN focuses solely on the three closest neighbors of each instance in the majority class and eliminates any instance that has a class differing from at least two of its three nearest neighbors. Conversely, NCL broadens its concept by implementing it within the minority class. The method operates by examining the neighbors of the minority class, identifying those with at least two out of three neighbors that differ in class labels, and subsequently, NCL eliminates the majority instances that are part of those nearest neighbors. The downsampling methods employed here utilize a data cleaning approach that focuses on eliminating less informative samples from the majority class, in contrast to the incremental addition of informative samples as proposed in this paper (Lee and Seo, 2022). This significant class imbalance prompted a problem in the experimental approach, leading to a new direction: clustering based on the specific type of cyber attack. This shift in focus aimed to address the imbalance by grouping similar attack patterns together, thereby facilitating a more nuanced analysis of the datasets. By clustering according to attack types, the experiments sought to uncover underlying patterns and distinctions within the minority class, ultimately enhancing the effectiveness of the analysis and classification processes. Although the data set's imbalance limited the use of clustering methods, trials on the complete dataset could be conducted to discover the optimal characteristics. The algorithms employed were Decision Tree, Random Forest, XGB, and Kneighbors (Figure 1, 2, 3, 4).

Figure 1
DecisionTree feature importance

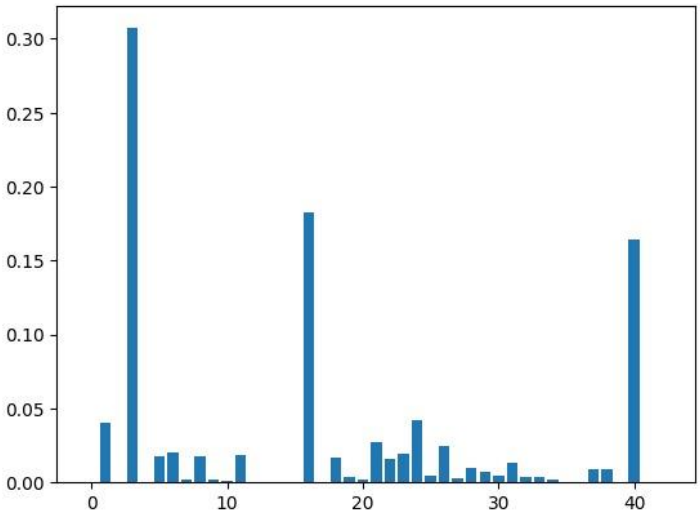
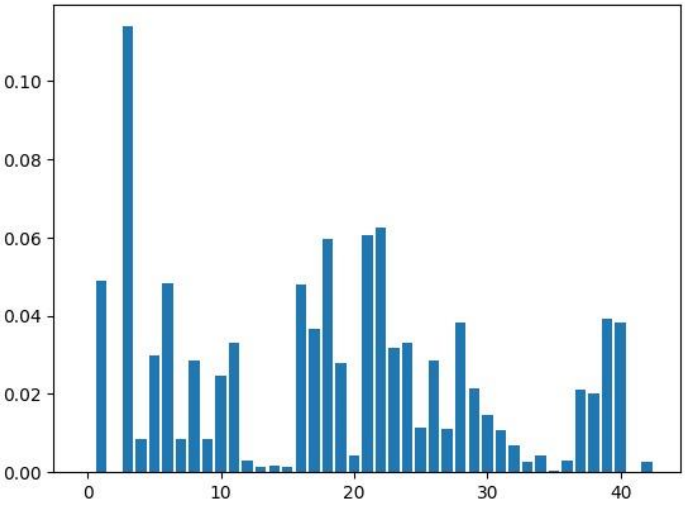


Figure 2
Random Forest feature importance



ISSN: 0067-3064

Figure 3
XGB fearure importance

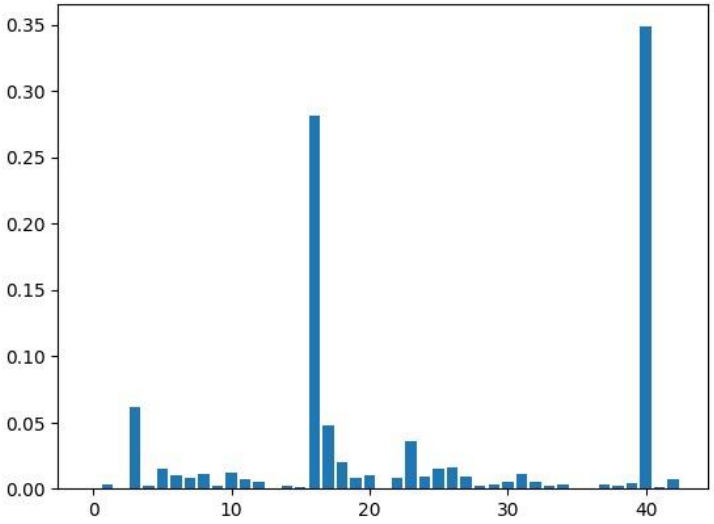
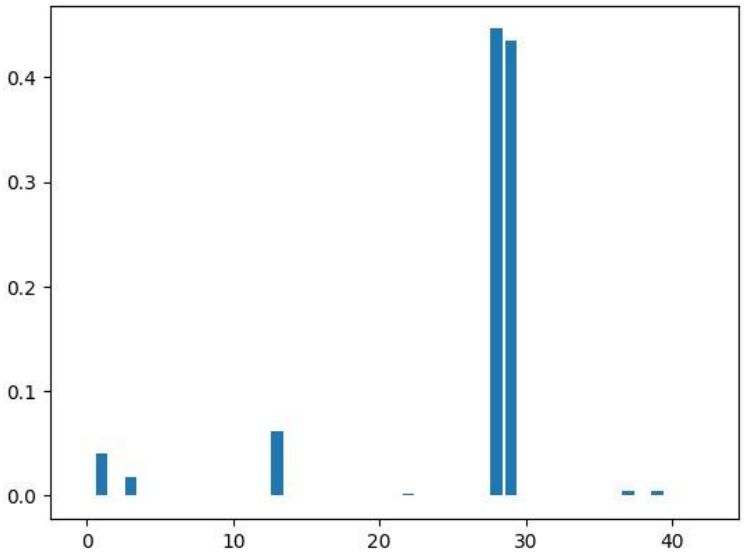


Figure 4
KNN feature importance



The decisions taken by the four classification algorithms reveal challenges that are already recognized within the domain of computer networks. The characteristic identified by the DecisionTree and RandomForest algorithms is positioned at level 4 within the OSI model. The fourth level of the Open Systems Interconnection (OSI) model is the transport layer, facilitating transparent data transfer between end users and

ISSN: 0067-3064

ensuring reliable data transfer services to the upper layers. The transport layer oversees the reliabiity of a particular link by implementing flow control, segmentation and reassembly, as well as error control mechanisms. Upon thorough examination of the dataset, it becomes evident that the attacks were executed across all three categories of ports at the OSI stack level 4. Within the framework of the OSI model, specifically at layer 4, known as the transport layer, ports are categorized into three primary types:

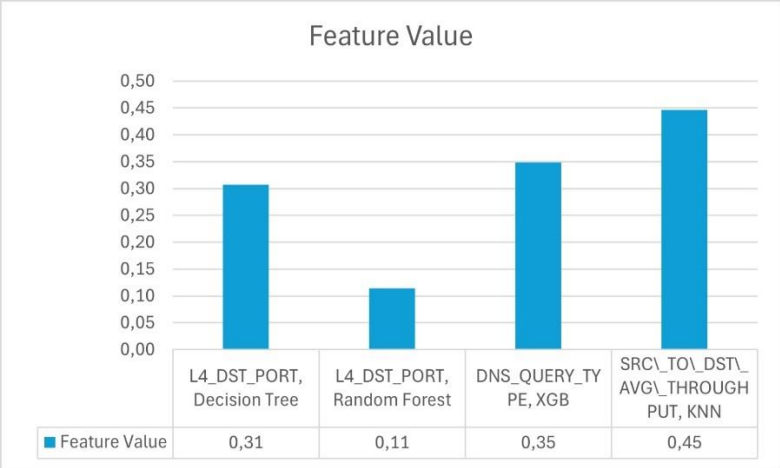
1. Well-Known Ports: These ports are designated for system operations or recognized services and protocols. Instances consist of HTTP (port 80), HTTPS (port 443), FTP (port 21), and SSH (port 22). The range of these ports extends from 0 to 1023.

2. Registered Ports: The Internet Assigned Numbers Authority (IANA) has registered these ports for designated services and applications. These are utilized by user processes or applications rather than by system processes. Examples of applications utilizing these ports include MySQL, which operates on port 3306, and Microsoft SQL Server, found on port 1433. The range of these ports is from 1024 to 49151.

3. Dynamic/Private Ports: The ports in question lack registration, allowing any application to utilize them for temporary communication purposes. The operating system allocates them dynamically for clientside communication as needed. These ports are referred to as ephemeral ports as well. The range of these ports extends from 49152 to 65535.

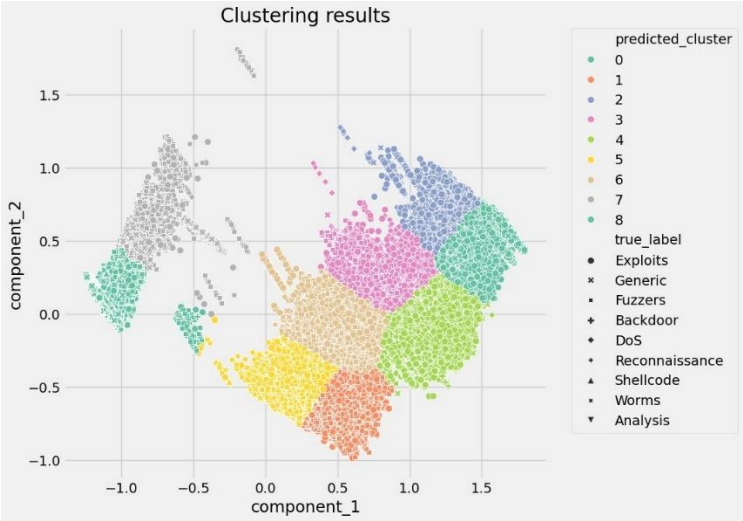
Among these three types of protocols, the ones that attackers most frequently utilize are those in the third category. Dynamic ports, often referred to as private ports, serve multiple purposes in hacking because of their adaptable and transient characteristics. An illustration of exploitation is dynamic port forwarding. This method facilitates the establishment of a secure tunnel connecting a local machine to a remote server, typically through SSH. This method can be utilized to circumvent network restrictions and gain access to internal services from an external network. Individuals with malicious intent can utilize a dynamic SOCKS proxy to channel traffic from various applications through an encrypted SSH tunnel, complicating detection efforts by network security teams. Another port recognized for its vulnerabilities is port 0. This falls into the initial category, and although this port is not officially recognized, packets are capable of being transmitted to and from port 0 on the internet. The creators of the original Berkeley UNIX ”Sockets” interface eliminated the specifications for port 0, allowing it to function as a wildcard. The zero port allows the operating system to autonomously allocate any other port it considers appropriate for the specific type of packet that requires Each algorithm classified the following feature as being the most important, we can see in Figure 5.

Figure 5
Features Values



To initiate clustering on the dataset, we need to preprocess the features. Specifically, we need to address the IPV4 SRC ADDR and IPV4 DST ADDR features. These features contain IP addresses formatted as X.X.X.X, where each X represents a number ranging from 0 to 255. To preserve the dataset's original structure, we opt not to utilize One Hot Encoder or Label Encoder. Instead, we simply remove the dot between the two numbers. The next step was to remove the rows from the data set that are labeled as Benign attack type, thus leaving only that 4% of the data set. The last step in preparing the data set was to remove the last column (the column that classifies the type of attack). Now we will create 2 pipelines. The first pipeline contains a scaler, namely MinMaxScaler and an algorithm for dimensionality reduction, PCA. PCA was chosen for dimensionality reduction because it is an unsupervised technique that aims to find the directions (principal components) that maximize the variance in the data. The 2nd pipeline contains the clustering algorithm, KMeans. The parameters used for the KMeans algorithm were the following: n clusters=n clusters (where n clusters represent the number of different attack types), init="k-means++", n init=50, max iter=500, random state=42. In Figure 6 is the result of running this Kmeans configuration on the dataset.

Figure 6
Kmeans Clustering Results



The silhouette score, when applied to preprocessed data with predicted labels, yielded a result of 0.564. From the Figure 6 we can see that different true labels (like Exploits, DoS) are appearing in the same predicted clusters (like 0 and 5), which could indicate that the clustering algorithm has difficulty separating these categories. This overlap could be due to the features used to create the cluster. We previously discussed the layer 4 destination port issue, but now let's look at the other two features: DNS query type and source to destination average throughput. A DNS query represents a request initiated by a user's device, such as a computer or mobile device, directed towards a DNS server to retrieve particular information. The main purpose is to identify the IP address linked to a specific domain name. Upon entering a website address in your browser, your device initiates a DNS query to convert the domain name into an IP address, a numerical identifier assigned to every device linked to a computer network. Domain names are userfriendly and memorable, yet the underlying technology depends on IP addresses for the identification and location of websites and services across the Internet. DNS queries can be categorized into three main types: recursive, iterative, and non-recursive queries. Each type operates differently in the DNS resolution process.

1. Recursive Query: This kind of query requires the DNS resolver to deliver a response. The resolver systematically queries multiple servers, beginning with the DNS Root Server and progressing to the Authoritative Name Server, in order to obtain the necessary information.

2. Iterative Query: In an iterative query, the DNS client solicits the resolver to deliver the most accurate response available. When the resolver possesses the information in its cache, it provides a direct response. In such cases, the client is

directed to a server that is geographically closer to the relevant DNS zone, prompting the client to reissue the query to this new server.

3. Non-Recursive Query: This method is employed when the resolver possesses the answer, either from its cache or by querying a DNS Name Server that is authoritative for the record. Additional query rounds, such as those found in recursive or iterative queries, are unnecessary.

If we were to look at our dataset, we could observe that the majority of DNS queries have a value of 0, which means NoError. The NoError response signifies that the executed DNS query returned a legitimate result. Nonetheless, this does not imply the absence of concerns. The NoError answer will not signify performance-related problems. It would be interesting to take a look in the future at the article written by (Ruan et al., 2013) called ”Pattern discovery in DNS query traffic”. They said that detecting anomalies in DNS query traffic is crucial for DNS service providers. Malfunctioning of the DNS server or network may lead to irregular query traffic and atypical or illicit behaviors among web users. Consequently, DNS query traffic reflects the condition of both the DNS and the network. To the best of our knowledge, the majority of current algorithms are either rulebased or blacklist-based, with rules that cannot be constantly modified. They provided an interesting topic, periodic query traffic trend mining, along with a method for identifying all periodic trend patterns inside a sequence database. They suggested that if a pattern manifests regularly in recent history, it is likely to recur with high probability until an anomalous event transpires. Given that patterns containing time intervals are often erratic and inconsistent, we focus solely on patterns devoid of intervals. They forecast the traffic volume for the subsequent moment by analyzing recent periodic query traffic trends. Network throughput measures the volume of data transmitted from one location to another over a defined timeframe.

The standard measurements are bits or bytes per second, denoted as bps, Kbps, Mbps, Gbps, or Tbps. If we take a look at our dataset, we can observe that in the case of traffic categorized as benign, the flow of information is relatively constant. In the case of attacks, the flow can vary from 1162512000 to 0.

Conclusion

The study emphasizes the critical role of machine learning and AI methods in classifying cyber threats. By employing techniques such as clustering and network feature analysis, hidden patterns and abnormal behaviors can be identified, enhancing the detection of attacks.

Challenges related to data imbalance, where the majority of samples are benign, were addressed by using clustering to group similar attacks. This enabled a more detailed analysis of minority classes, providing valuable insights into how attacks are distributed within the dataset.

The experiments highlighted difficulties in separating similar attack types, such as Exploits and DoS, due to shared characteristics. This underscores the need to combine clustering with other methods, such as classification-based analysis or dimensionality reduction.

Features such as destination ports at Layer 4 of the OSI model, DNS query types, and data throughput proved critical. These characteristics illustrate how cyberattacks exploit vulnerabilities in network infrastructure.

The use of Principal Component Analysis (PCA) for dimensionality reduction improved the efficiency of the clustering process. This suggests that unsupervised preprocessing techniques can enhance the analysis of complex network data.

While clustering produced promising results, the conclusions highlight the necessity of a hybrid approach that combines supervised and unsupervised techniques for more accurate classification. Adaptability of methods is essential to address the constantly evolving nature of cyber threats. In the future we intend to explore more sophisticated clustering techniques (DBSCAN, Gaussian Mixture Model, BIRCH, Affinity Propagation), and we also aim to construct a classification approach utilizing Convolutional Neural Networks (CNN).

These conclusions point to future research directions, focusing on developing more robust and integrated methods for detecting and preventing cyber threats.

Conceptualization, M. V. C. and A. M. C.; methodology M. V. C. and A. M. C.; software, M. V. C. and A. M. C.; validation M. V. C. and A. M. C.; formal analysis, M. V. C. and A. M. C.; investigation, M. V. C. and A. M. C.; resources M. V. C. and A. M. C.; data curation, M. V. C. and A. M. C.; writing—original draft preparation, M. V. C. and A. M. C.; writing—review and editing, M. V. C. and A. M. C.; visualization, M. V. C. and A. M. C.; supervision, M. V. C. and A. M. C.; project administration, M. V. C. and A. M. C. All authors have read and agreed to the published version of the manuscript.

Funding: This study did not need external funding to be carried out.

Conflicts of Interest: The authors of this manuscript declare that they have no affiliations or in-volvement with any organization or entity that has a financial interest or non-financial interest in the subject matter or materials discussed. Financial interests include honoraria, educational grants, participation in speakers' bureaus, membership, employment, consultancies, stock ownership, eq-uity interest, expert testimony, or patent-licensing arrangements. Non-financial interests include personal or professional relationships, affiliations, knowledge, or beliefs.

References

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150. https://doi.org/10.1002/ett.4150

Alduailij, M., Khan, Q. W., Tahir, M., Sardaraz, M., Alduailij, M., & Malik, F. (2022). Machine-learning-based DDoS attack detection using mutual information and random forest feature importance method. *Symmetry*, *14*(6), 1095. https://doi.org/10.3390/sym14061095

Aung, Y. Y., & Min, M. M. (2018). An analysis of K-means algorithm based network intrusion detection system. *Advances in Science, Technology and Engineering Systems Journal*, *3*(1), 496-501. 10.25046/aj030160

Ben Jabeur, S., Stef, N., & Carmona, P. (2023). Bankruptcy prediction using the XGBoost algorithm and variable importance feature engineering. *Computational Economics*, *61*(2), 715-741. https://doi.org/10.1007/s10614-021-10227-1

Bohara, B., Bhuyan, J., Wu, F., & Ding, J. (2020). A survey on the use of data clustering for intrusion detection system in cybersecurity. *International journal of network security & its applications*, *12*(1), 1. https://doi.org/10.5121/ijnsa.2020.12101

Habib, A. B. (2021). Elbow method vs silhouette Co-efficient in determining the number of clusters. *Rapp. tech. June*. 10.13140/RG.2.2.27982.79688

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, *622*, 178-210. https://doi.org/10.1016/j.ins.2022.11.139

Lee, W., & Seo, K. (2022). Downsampling for binary classification with a highly imbalanced dataset using active learning. *Big Data Research*, *28*, 100314. https://doi.org/10.1016/j.bdr.2022.100314

Liu, H., Chen, J., Dy, J., & Fu, Y. (2023). Transforming complex problems into K-means solutions. *IEEE transactions on pattern analysis and machine intelligence*, *45*(7), 9149-9168.

Magán-Carrión, R., Urda, D., Díaz-Cano, I., & Dorronsoro, B. (2020). Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *Applied Sciences*, *10*(5), 1775. https://doi.org/10.3390/app10051775

Nowak, J., Korytkowski, M., Wozniak, M., & Scherer, R. (2019). URL-based Phishing Attack Detection by Convolutional Neural Networks. *Aust. J. Intell. Inf. Process. Syst.*, *15*(2), 60-67.

Portnoy, L. (2000). *Intrusion detection with unlabeled data using clustering* (Doctoral dissertation, Columbia University).

Ruan, W., Liu, Y., & Zhao, R. (2013). Pattern discovery in DNS query traffic. *Procedia Computer Science*, *17*, 80-87. https://doi.org/10.1016/j.procs.2013.05.012

Rusland, N. F., Wahid, N., Kasim, S., & Hafit, H. (2017, August). Analysis of Naïve Bayes algorithm for email spam filtering across multiple datasets. In *IOP conference*

series: materials science and engineering (Vol. 226, No. 1, p. 012091). IOP Publishing. 10.1088/1757-899X/226/1/012091

Sarhan, M., Layeghy, S., & Portmann, M. (2022). Towards a standard feature set for network intrusion detection system datasets. *Mobile networks and applications*, 1-14. https://doi.org/10.1007/s11036-021-01843-0

Susilo, B., & Sari, R. F. (2020). Intrusion detection in IoT networks using deep learning algorithm. *Information*, *11*(5), 279. https://doi.org/10.3390/info11050279

Wei, W., Ke, Q., Nowak, J., Korytkowski, M., Scherer, R., & Woźniak, M. (2020). Accurate and fast URL phishing detector: a convolutional neural network approach. *Computer Networks*, *178*, 107275. https://doi.org/10.1016/j.comnet.2020.107275

Yuan, C., & Yang, H. (2019). Research on K-value selection method of K-means clustering algorithm. *J*, *2*(2), 226-235. https://doi.org/10.3390/j2020016

Alam, K., Monir, M. F., Hassan, Z., & Habib, M. T. (2024, October). Optimizing IoT Network Intrusion Detection: A Deep Learning Approach. In *2024 7th Conference on Cloud and Internet of Things (CIoT)* (pp. 1-5). IEEE.

Bhuiyan, M. H., Alam, K., Shahin, K. I., & Farid, D. M. (2024, September). A Deep Learning Approach for Network Intrusion Classification. In *2024 IEEE Region 10 Symposium (TENSYMP)* (pp. 1-6). IEEE.

Ba, A., & Adda, M. (2024). Intrusion Detection in IIoT Using Machine Learning. *Procedia Computer Science*, *251*, 265-272.

Yang, C., Wu, L., Xu, J., Ren, Y., Tian, B., & Wei, Z. (2024). Graph Learning Framework for Data Link Anomaly Detection. IEEE Access.