MJFAS
Malaysian Journal of Fundamental and Applied Sciences

RESEARCH ARTICLE

Comparative Analysis of Machine Learning Models to Predict Common Vulnerabilities and Exposure

Shaesta Khan Sheh Rahman[a], Noraziah Adzhar[a*], Nazri Ahmad Zamani[b]

[a]Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan, Pahang, Malaysia; [b]Cyber Threat Intelligence Department, Cybersecurity Malaysia, Menara Cyber Axis, Jalan Impact, 63000 Cyberjaya, Selangor, Malaysia

Abstract Predicting Common Vulnerabilities and Exposures (CVE) is a challenging task due to the increasing complexity of cyberattacks and the vast amount of threat data available. Effective prediction models are crucial for enabling cybersecurity teams to respond quickly and prevent potential exploits. This study aims to provide a comparative analysis of machine learning techniques for CVE prediction to enhance proactive vulnerability management and strengthening cybersecurity practices. The supervised machine learning model which is Gaussian Naive Bayes and unsupervised machine learning models that utilize clustering algorithms which are K-means and DBSCAN were employed for the predictive modelling. The performance of these models was compared using performance metrics such as accuracy, precision, recall, and F1-score. Among these models, the Gaussian Naive Bayes achieved an accuracy rate of 99.79%, and outperformed the clustering-based machine learning models in effectively determining the class labels or results of the data it was trained on or tested against. The outcome of this study will provide a proof of concept to Cybersecurity Malaysia, offering insights into the CVE model.

Keywords: Cyber threat, common vulnerabilities and exposures, unsupervised and supervised machine learning models, accuracy.

Introduction

Cyber threats are sporadic and not limited to governments but also companies and individuals [1,2]. Growing threats have been identified in emerging technologies such as social media, cloud computing, web applications, and smartphone technologies [3,4]. The rise of cyber threat incidents highlights the urgent need for vulnerability management and effective mitigation strategies. Vulnerabilities are weaknesses that attackers exploit to access and conduct illegitimate activities unlawfully [5]. This includes executing code, installing various types of malwares [6], acquire, modify, or even destroy sensitive data. The most recent threat landscape demonstrates how tough it is to stop an incident since attackers can aim weaknesses in people, procedures, and technology [7]. This is due to advancements in hackers' strategies and tactics, which have become increasingly difficult to detect, investigate and resolve. In [7] it also said that the organized crime groups that use ransomware to encrypt vital data and systems have an impact on a lot of businesses.

In cybersecurity, the Common Vulnerabilities and Exposures (CVE) initiative by MITRE Corporation [8], initiated in 1999, provides a framework for detecting and classifying vulnerabilities. The Common Weakness Enumeration (CWE) system further assists by identifying, categorizing, and explaining common software problems, acting as a tool for addressing flaws. As the field of cybersecurity continues to advance, so does the severity of cyber threats. Attackers employ increasingly sophisticated methods to compromise systems, escalating the stakes. The prevailing reactive approach to cyberattack response, which addresses threats only after systems have been breached, is no longer sufficient. Detecting concealed threats in vast, complex environments is challenging, rendering absolute perimeter defence impractical.

***For correspondence:**
noraziahadzhar@umpsa.edu.my

Received: 27 Aug. 2024
Accepted: 04 Nov. 2024

©Copyright Sheh Rahman. This article is distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use and redistribution provided that the original author and source are credited.

e-ISSN 2289-599X | DOI: https://doi.org/10.11113/mjfas.v20n6.3822

In [9] state that machine learning (ML) approaches can effectively analyze big datasets and identify hidden patterns related to current and future risks. Various ML algorithms can also be used in cybersecurity to perform spam detection, virus detection, denial-of-service attacks, and network anomaly detection including supervised and unsupervised machine learning models [10,11,12]. Naive Bayes is one of the popular technique and performed well in many practical applications such as text classification, spam filtering and cyber threat detection [13,14,15]. The strength of this method is due to its simple construction and efficient in both learning and classification task [11]. Naive Bayes is used in [16], a study in solving anomaly detection problem. From the study, it is reported that Naive Bayes model works effectively by producing high accuracy in most categories tested. In [17], this study also employs a simple framework of Naïve Bayes model in network intrusion detection. The KDD'99 dataset is used for training and testing. The result from this study shows that the classifier achieved 96%, 99%, 90% and 90% testing accuracy for all four groups of data formed. It can be inferred that Naive Bayes finds a good amount of use in cybersecurity. A study in [18] evaluated the effectiveness of fundamental machine learning classifiers, using metrics such as precision, recall, and F1-score. The studies shown consistent results across different vulnerability types, demonstrating machine learning's viability for automated vulnerability classification.

In this study, we are interested in providing a comparative analysis between the supervised and unsupervised machine learning models, specifically the clustering-based approach, to compare their effectiveness in predicting CVE. From previous studies, it can be seen that Naive Bayes algorithm is one of the powerful tools and can be very competitive. Therefore, we would like to employ this supervised machine learning model specifically the variant of Gaussian Naïve Bayes model. In order to evaluate the performance of this model, the results were compared with K-means [19] and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) model [20], the clustering-based unsupervised machine learning model, considered as one of the well-known learning techniques. By conducting modeling tests and evaluating metrics such as accuracy, precision, recall, and F1-score, this study aims to enhance vulnerability management and improve the timely detection, correction, and prevention of security vulnerabilities.

The remaining part of this paper is as follows: Section 2 explains the research methodology stages and data engineering process. This section also covers exploratory data analysis and the results obtained from the machine learning algorithm for predicting cyber threats. Finally, Section 3 summarizes this paper.

Research Methodology

This section provides an overview of the research methodology stages of this study. The whole process is divided into five key steps. The first step is business understanding which involves defining the research problem, performing a literature review and outlining the objectives. The second step is data understanding which is to perform data collection procedure and possible data preparation process. In the third step, exploratory data analysis was conducted to understand the data's characteristics and gain insights into the problem. In the next step, machine learning techniques which are K-means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Gaussian Naive Bayes were implemented for CVE prediction. Lastly, the performance for each method were evaluated in based on the following metrics: accuracy, precision, recall, and F1 score. In order to successfully achieve the study's objectives, the steps outlined must be followed, as illustrated in Figure 1.

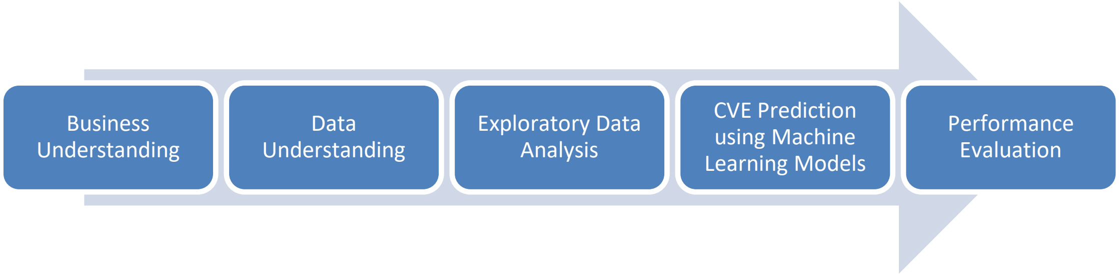

Business Understanding
Data Understanding
Exploratory Data Analysis
CVE Prediction using Machine Learning Models
Performance Evaluation

Figure 1. Research plan flowchart

Business Understanding

The field of cybersecurity has seen significant advancements in recent years, yet cyber threats have grown more severe. This escalation is due to the increasingly sophisticated techniques used by attackers to compromise organizational systems. The stakes are rising, and current cybersecurity responses remain predominantly reactive or defensive, meaning threats are often addressed only after systems have already been breached. The main objective in this study is to perform comparative study among Machine Learning algorithms to identify an effective strategy in predicting the CVE. Prediction models involved in this study are the supervised machine learning algorithm, Gaussian Naive Bayes and two clustering-based unsupervised machine learning algorithms which are K-means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Performance metric such as accuracy, precision, recall and F1-score will be utilized to assess the performance for each model. By examining and comparing the predictive capabilities of these methods, the study hope to contribute to a proactive vulnerability management and the advancement of cybersecurity practices. The implementation of this architecture leverages the Python programming language for modelling tasks, utilizing popular libraries and frameworks for machine learning and data analysis such as NumPy, Pandas, Scikit-learn, seaborn and matplotlib.

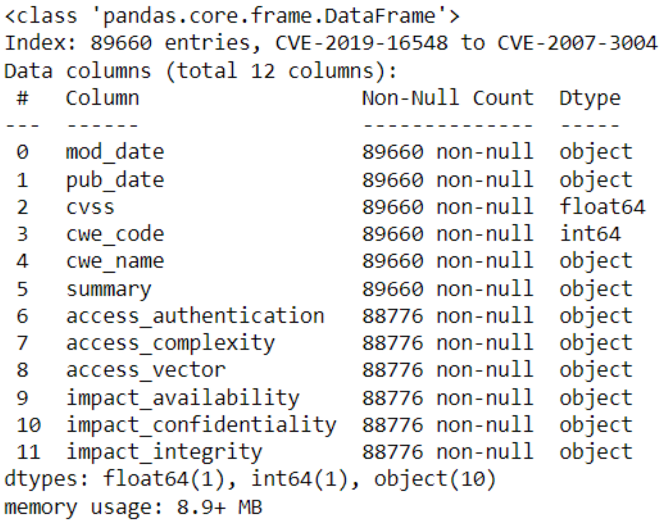
Data Understanding

The dataset used in this study is the CVE dataset obtained from Kaggle. This dataset, which spans from 1999 to 2019, was sourced from the National Cyber Security Division of the United States Department of Homeland Security (NIST). It includes detailed information on cybersecurity threats, such as descriptions of vulnerabilities, impacted components, severity scores, and references to mitigation measures. The data is drawn from authoritative sources, including the U.S. Department of Homeland Security and the MITRE Corporation, and is provided in CSV format. The initial step in data preparation involves importing the necessary library in Python such as in Table 1.

Table 1. Python libraries imported and their purpose

Python Library	Purpose
NumPy	Numerical computing
pandas	Data processing
Scikit-learn	Predictive data analysis
seaborn	Data visualization
Matplotlib	Data visualization

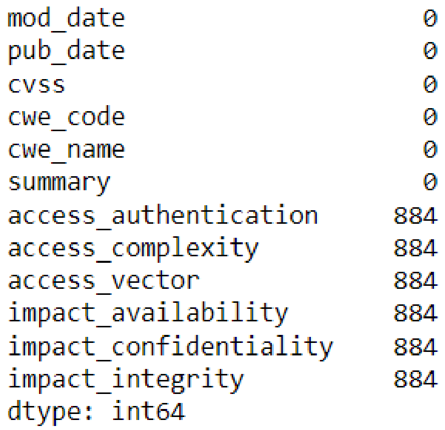
The dataset was then imported into Python, and a data description was obtained. As shown in Figure 2, the dataset contains 12 features with 89,660 records. The data type for each feature was also determined.



```
<class 'pandas.core.frame.DataFrame'>
Index: 89660 entries, CVE-2019-16548 to CVE-2007-3004
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   mod_date                89660 non-null  object
 1   pub_date                89660 non-null  object
 2   cvss                    89660 non-null  float64
 3   cwe_code                89660 non-null  int64
 4   cwe_name                89660 non-null  object
 5   summary                 89660 non-null  object
 6   access_authentication   88776 non-null  object
 7   access_complexity       88776 non-null  object
 8   access_vector           88776 non-null  object
 9   impact_availability     88776 non-null  object
 10  impact_confidentiality  88776 non-null  object
 11  impact_integrity        88776 non-null  object
dtypes: float64(1), int64(1), object(10)
memory usage: 8.9+ MB
```

Figure 2. Features and data type

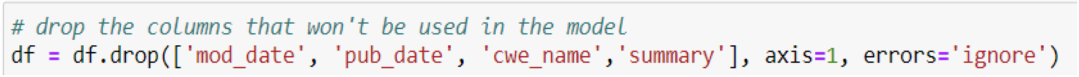
Before proceeding with data analysis, the dataset needs to undergo data cleaning process to identify inconsistencies within the data. A clean dataset can improve overall productivity, consistency, and reliability of the data for analysis or modelling purposes. One of critical task in data cleaning is to check for null or missing values to ensure data completeness. In this study, all missing values are represented as NaN (Not a Number), which indicates the absence of value for a specific variable. The output of using Pandas' `.isnull()` function to detect missing values in the dataset is shown in Figure 3.



```
mod_date                   0
pub_date                   0
cvss                       0
cwe_code                   0
cwe_name                   0
summary                    0
access_authentication    884
access_complexity        884
access_vector            884
impact_availability      884
impact_confidentiality   884
impact_integrity         884
dtype: int64
```

Figure 3. Output for missing values

Then, the query to replace or impute the missing values in the dataset was executed. The specific approach for filling in the missing values will depend on the nature of the data and the context of the analysis. In this study, each missing values was imputed with zero values. This choice was driven by the categorical nature of the columns in the dataset, which poses challenges in utilizing measures such as mean, median, and mode due to the specific characteristics of these columns. Finally, the unwanted column will be removed. The act of removing these columns serves to simplify the dataset and enhance the accuracy of the analysis. Figure 4 visually represents the query used to drop the unwanted column.



```
# drop the columns that won't be used in the model
df = df.drop(['mod_date', 'pub_date', 'cwe_name','summary'], axis=1, errors='ignore')
```

Figure 4. Query of removing unwanted columns

The columns that are being removed are those that lack relevance or utility in the analysis or modelling process. By removing this unwanted columns, the dataset's dimensionality is reduced, leading to improved computational efficiency and a lower risk of overfitting. Therefore, the removal of unwanted columns is a crucial step in preparing the dataset for analysis or modelling purposes. The dataset was then processed using one-hot encoding and label encoding to convert the data into a compatible format for machine learning algorithms.

Exploratory Data Analysis

Before proceeding to modeling, an exploratory data analysis was conducted to gain insights into the dataset. Exploratory data analysis will provide insights and help to uncover the pattern and trend of the threat event [21]. Figure 5 visualizes the number of threats based on their publication dates.

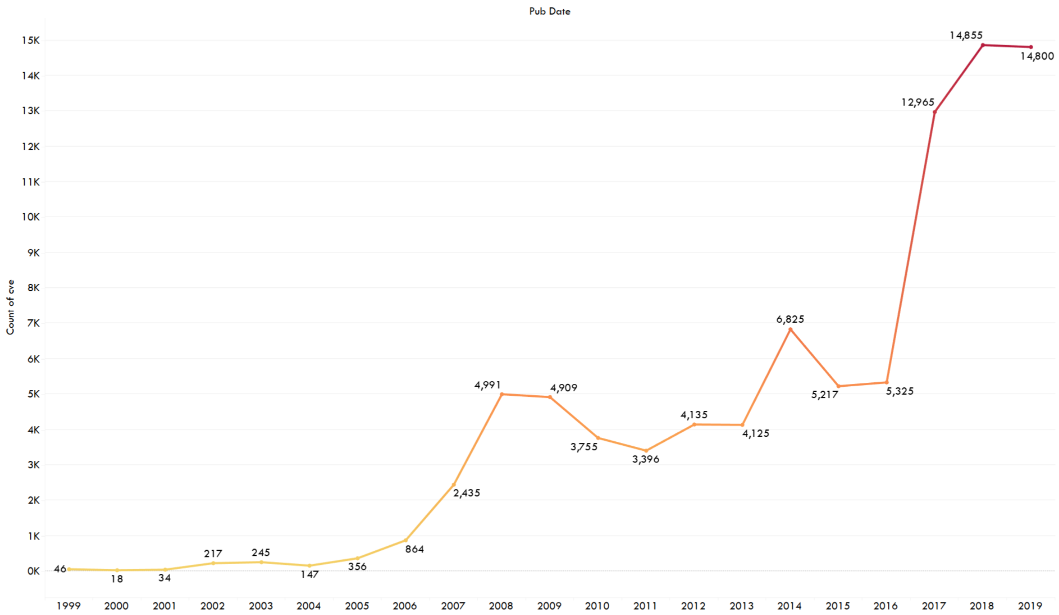


Figure 5. Number of threats by year

The color of the line graph in Figure 5 indicates the number of threats that happened across the year. The dark line represents the increase in CVE while gradually brighter as the number decreases. The graph also reveals some fascinating patterns. It shows that CVE experienced a consistent upward trend from 1999 to 2003, followed by a slight dip in 2004. However, the upward trajectory resumed and continued until 2008. This cyclical rise and fall pattern persisted, indicating a recurring trend. Interestingly, between 2016 and 2019, there was a remarkable surge in CVE, suggesting a significant increase during that period. These findings, allowing one to grasp the dynamic nature of CVE occurrences throughout time. Following this, an analysis of the Common Weakness Enumeration (CWE) based on the publication date is conducted. Table 2 shows the top 10 CWE categories for further analysis.

Table 2. Top 10 CWE category

CWE category
Cryptographic Issues
Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
Improper Control of Generation of Code ('Code Injection')
Resource Management Errors
Information Exposure
Permissions Privileges and Access Controls
Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
Improper Input Validation
Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
Improper Restriction of Operations within the Bounds of a Memory Buffer

The number of cases recorded for each of these CWE categories are then visualized through a line graph as illustrated in Figure 6 where each colour represents a different CWE category.

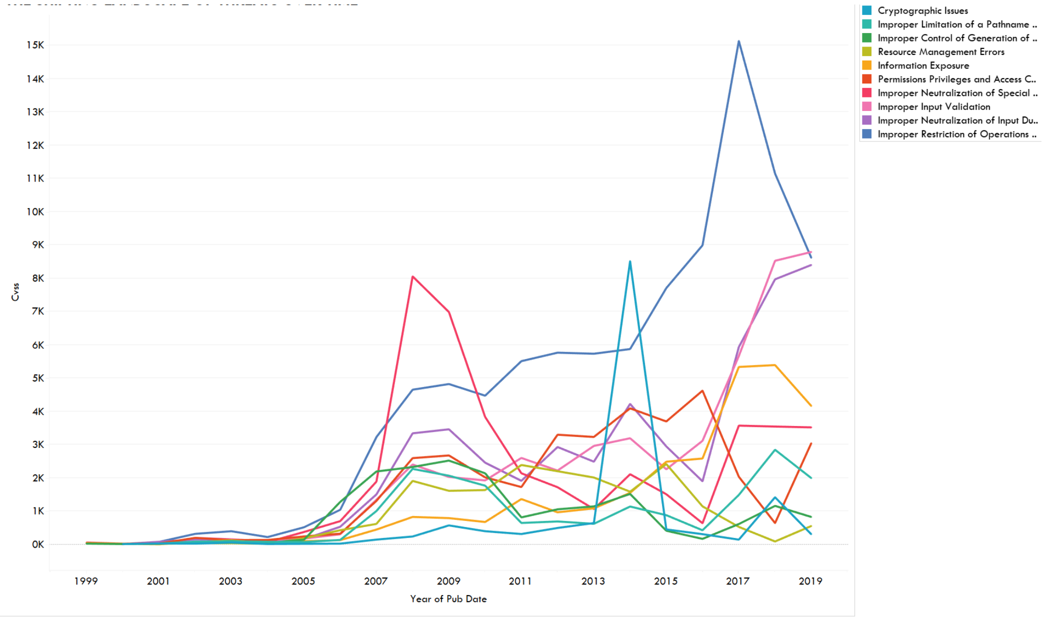


Figure 6. The shifting landscape of threat over year

The line graph in Figure 6 showcases the top 10 frequently encountered vulnerabilities. For most categories, there is a general increase in the number of cases starting around 2006, peaking around 2016, and then decline thereafter. The major peak records “Improper Neutralization of Input During Web Page Generation” as the highest, reaching over 15000 cases. Another peak that can be seen is the “Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')” around 2008, “Cryptographic Issues” in 2014, and “Improper Input Validation” around 2017. After 2016, many categories exhibit a significant decrease in cases. This suggests that effective efforts to mitigate the issues may have been implemented.

On the other hand, there is a rising prevalence of “Improper Input Validation” and “Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')” vulnerabilities. This implies a growing need to address these weaknesses in software systems. By closely monitoring these trends and adapting security measures accordingly, developers can effectively prioritize their efforts and enhance the overall security of their software applications.

CVE Prediction Using Machine Learning Models

Following the research plan in Figure 1, this phase aimed to provide the comparative analysis of machine learning (ML) algorithms in predicting the Common Vulnerabilities and Exposure (CVE). This study explored various machine learning methods to compare their performance, which are Gaussian Naïve Bayes (supervised ML model) and clustering-based unsupervised models, K-means and DBSCAN. The dataset training set is utilized with a 70-30 split ratio. The models were implemented with the principal objective of assessing their prediction accuracy and identifying the method with the highest accuracy.

Gaussian Naive Bayes

Naive Bayes is a machine learning algorithm that utilizes probability and is commonly used for classification tasks. Gaussian Naive Bayes, a variant of Naive Bayes algorithm, assumes that the data follows a Gaussian or normal distribution. This assumption simplifies implementation by requiring only the calculation of the mean and standard deviation of the training data, rather than estimating the entire data distribution with more complex functions. The `var_smoothing` hyperparameter in Gaussian Naïve Bayes adds a small value to the variances of all features, preventing numerical instability when dealing with features that have zero variance. Hyperparameter tuning identified 1e-07 as the optimal value for `var_smoothing` based on the evaluation criteria used. The accuracy achieved with these optimal hyperparameters is 0.9979, meaning the algorithm correctly classified approximately 99.79% of the cases in the dataset. This exceptional result indicates that the algorithm is highly effective at differentiating between classes and making accurate predictions.

K-Means Method

K-means is one of the most widely used and simplest unsupervised algorithms for solving clustering problems. It involved classifying a given dataset into a predetermined number of clusters, often denoted as k clusters. The optimal number of clusters can be determined using the elbow method by calculating the Within-Cluster Sum of Squares (WCSS):

$$\sum_{i=1}^{K} \sum_{x \in C_i} \left\| x_i - \mu_i \right\|^2 \qquad (1)$$

where

K : number of clusters,

x_i : data point in cluster C_i ,

μ_i : centroid of the cluster C_i , and

$\left\| x_i - \mu_i \right\|$: distance between a data point x_i and its centroid μ_i

Figure 7 shows the result from the elbow method on the trained dataset.

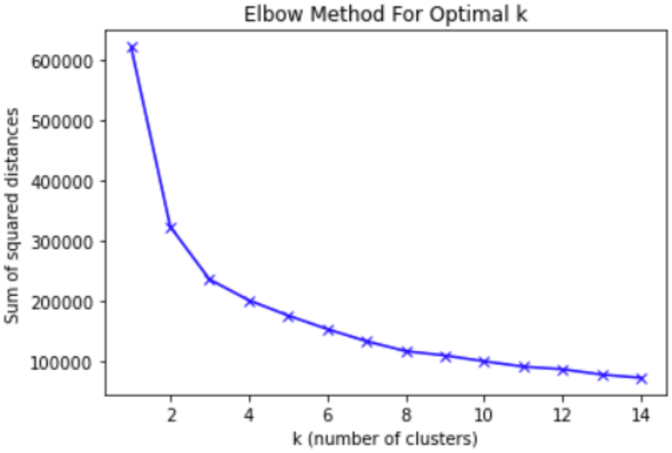


Figure 7. Result from Elbow Method

Based on the findings in Figure 7, the optimal number of clusters, k is determined to be 3. With this value, the K-means algorithm is applied, and the silhouette score is calculated to evaluate the quality of the resulting clusters. The silhouette score is 0.4604 indicating a moderate level of separation and compactness among the clusters. This suggests that while K-means has effectively grouped the data points, some overlap or ambiguity between clusters may still exist.

The accuracy score for the K-means algorithm in this study is 0.2674, indicating that the algorithm correctly classified approximately 26.74% of the instances in the dataset. This score reflects the relatively low performance of the algorithm in accurately classifying the data.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is an unsupervised learning algorithm that clusters data points based on their density. The key steps in DBSCAN approach include data preparation, selecting parameters such as epsilon (ε) and minimum points, identifying core points, expanding clusters, detecting noise points, assigning data points to clusters, and evaluating the clustering results.

The success of DBSCAN relies heavily on proper dataset preparation and careful selection of parameters. Core points are identified based on their neighbourhood density, and clusters are expanded by including density-reachable points. Noise points, which do not belong to any cluster, are also detected. Data points are then assigned to clusters based on their connectivity. The quality of clustering can be assessed using metrics like the silhouette coefficient or through visual inspection.

In this study, the dataset shows a silhouette score of 0.9971, indicating that the clusters formed by DBSCAN are highly distinct and well-separated. The data points within each cluster are tightly grouped, while the clusters themselves are well separated, suggesting that DBSCAN has successfully identified meaningful patterns in the data. However, the accuracy score is 0.4373, meaning that the algorithm correctly predicted the class labels for approximately 43.73% of the instances in the dataset. While this accuracy score is not particularly high, it is important to consider the dataset's context and characteristics when interpreting this result.

Performance Evaluation

Evaluation in machine learning refers to the process of assessing the performance and quality of a machine learning model or algorithm. It involves measuring how effectively the model can make predictions or classifications on unseen data. This process is crucial for selecting the best model among different algorithms. In this study, Accuracy, Precision, Recall, and F1-score were considered for model evaluation and the respective formulas are shown below.

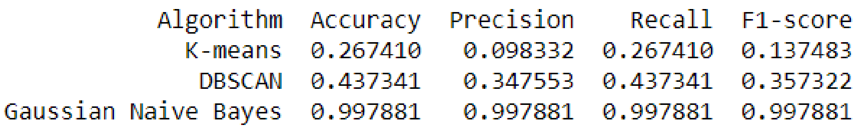
$$\text{Accuracy} = \frac{TruePositive(TP) + TrueNegative(TN)}{TruePositive(TP) + TrueNegative(TN) + FalsePositive(FP) + FalseNegative(FN)} \quad (2)$$

$$\text{Precision} = \frac{TruePositive(TP)}{TruePositive(TP) + FalsePositive(FP)} \quad (3)$$

$$\text{Recall} = \frac{TruePositive(TP)}{TruePositive(TP) + FalseNegative(FN)} \quad (4)$$

$$\text{F1-score} = \frac{2\text{*Precision*Recall}}{\text{Precision+Recall}} \quad (5)$$

Figure 8 presents the performance metrics for the three different algorithms K-means, DBSCAN, and Gaussian Naïve Bayes across four evaluation metrics: Accuracy, Precision, Recall, and F1-score while Figure 9 visualizes the evaluation metrics for each machine learning algorithm on the test data set.



```
           Algorithm  Accuracy  Precision    Recall  F1-score
             K-means  0.267410   0.098332  0.267410  0.137483
              DBSCAN  0.437341   0.347553  0.437341  0.357322
Gaussian Naive Bayes  0.997881   0.997881  0.997881  0.997881
```

Figure 8. Comparison of evaluation for each model

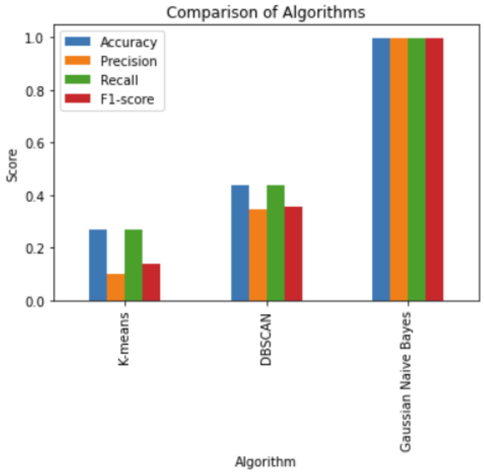


Figure 9. Plot showing performance evaluation on various techniques

From the results, it can be seen that K-means performs poorly across all metrics, with particularly low precision (9.83%) and F1-score (13.75%), indicating that it struggles significantly to correctly identify and classify instances. Its accuracy and recall are also low, making it the least effective algorithm among the three. DBSCAN shows moderate performance, with accuracy and recall both at 43.73%. Precision and F1-score are somewhat lower, suggesting that while DBSCAN is better than K-means, it still has limitations in classification tasks, particularly in balancing precision and recall. Gaussian Naïve Bayes significantly outperforms the other two algorithms, with near-perfect scores across all metrics. Its accuracy, precision, recall, and F1-score are all 99.79%, indicating that it is extremely effective at correctly identifying and classifying instances. This result is consistent with previously reported studies in the literature [13,14]

In summary, Gaussian Naive Bayes is the most reliable and accurate algorithm among the three, with nearly perfect performance. DBSCAN provides moderate results but is considerably less effective than Gaussian Naive Bayes. K-means is the weakest algorithm in this study, with very low scores across all evaluation metrics. This supports that a supervised machine learning is more effective in detecting CVE due to its ability to leverage labelled data. Without labeled data, it will be challenging for unsupervised machine learning models to differentiate patterns often lead to high false positives or false negatives. For future direction of this research, new learning strategies are needed to improve the robustness of the unsupervised machine learning model.

Conclusions

In conclusion, the Common Vulnerabilities and Exposures (CVE) framework is crucial for maintaining up-to-date cybersecurity techniques, especially in the context of the fourth industrial revolution and the widespread use of Internet of Things (IoT) devices. This study successfully implemented data engineering processes, including data cleaning, one-hot encoding, and label encoding, to prepare and optimize data for analysis. Exploratory Data Analysis (EDA) provides valuable insights into CVE patterns, revealing significant trends, such as the surge in incidents in 2019 and the vulnerability of operating system products. This study provides a comparative analysis of the supervised machine learning model and two clustering-based machine learning models. The Gaussian Naive Bayes algorithm emerged as the most effective predictive model, achieving an accuracy of 99.79%, significantly outperforming K-means and the DBSCAN algorithm. Additionally, in terms of precision, recall, and F1-score, Gaussian Naïve Bayes consistently outperformed the other models, earning near-perfect scores, indicating an outstanding capacity to accurately identify events. These findings shows that supervised machine learning or semi-supervised machine learning is more effective in CVE detection due to their ability to leverage labeled data.These findings underscore the importance of cybersecurity awareness, proactive vulnerability management, and the practical application of Gaussian Naive Bayes, offering valuable insights for organizations and researchers in the cybersecurity field, particularly Cybersecurity Malaysia.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Acknowledgement

The authors express their gratitude for the support provided by the Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, and Cybersecurity Malaysia. Appreciation is also extended to the reviewers for their valuable comments and suggestions on this paper.

References

[1] Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Reports*, *7*, 8176–8186. https://doi.org/10.1016/j.egyr.2021.08.126

[2] Ophoff, J., & Berndt, A. (2020). Exploring the value of a cyber threat intelligence function in an organization. In *IFIP advances in information and communication technology* (pp. 96–109). https://doi.org/10.1007/978-3-030-59291-2_7

[3] Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, *80*(5), 973–993. https://doi.org/10.1016/j.jcss.2014.02.005

[4] Sarker, I.H., Kayes, A.S.M., Badsha, S. *et al.* Cybersecurity data science: an overview from machine learning perspective. *J Big Data* **7**, 41 (2020). https://doi.org/10.1186/s40537-020-00318-5

[5] Leverett, É., Rhode, M., & Wedgbury, A. (2021). Vulnerability Forecasting: Theory and practice. *Digital Threats Research and Practice*, *3*(4), 1–27. https://doi.org/10.1145/3492328

[6] Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics, 12(6), 1333. https://doi.org/10.3390/electronics12061333

[7] Sahrom Abu, M., Rahayu Selamat, S., Ariffin, A., & Yusof, R. (2018). Cyber Threat Intelligence – Issue and Challenges. *Indonesian Journal of Electrical Engineering and Computer Science*, *10*(1), 371. https://doi.org/10.11591/ijeecs.v10.i1.pp371-379

[8] Grigorescu, O., Nica, A. A., Dascalu, M., & Rughinis, R. (2022). CVE2ATT&CK: BERT-Based Mapping of CVEs to MITRE ATT&CK Techniques. *Algorithms*, *15*(9), 314. https://doi.org/10.3390/a15090314

[9] Yeboah-Ofori, A., Ismail, U. M., Swidurski, T., & Opoku-Boateng, F. (2021). Cyberattack Ontology: A Knowledge Representation for Cyber Supply Chain Security. *2021 International Conference on Computing, Computational Modelling and Applications (ICCMA)*. https://doi.org/10.1109/iccma53594.2021.00019

[10] Tang, X., Astle, Y. S., & Freeman, C. (2020). Deep Anomaly Detection with Ensemble-Based Active Learning. *2020 IEEE International Conference on Big Data (Big Data)*. https://doi.org/10.1109/bigdata50022.2020.9378315

[11] Ahsan, M., Nygard, K. E., Gomes, R., Chowdhury, M. M., Rifat, N., & Connolly, J. F. (2022). Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review. *Journal of Cybersecurity and Pivacy*, *2*(3), 527–555. https://doi.org/10.3390/jcp2030027

[12] Naseer, I. (2024). Machine Learning Applications in Cyber Threat Intelligence: A Comprehensive review. *Deleted Journal*, *3*(2), 190–200. https://doi.org/10.62019/abbdm.v3i2.85

[13] Setiadi F. F.,Kesiman M. W. A. and Aryanto 2021 Detection of dos attacks using naive bayes method based on internet of things (iot)J. Phys.: Conf. Ser. doi: 10.1088/1742-6596/1810/1/012013

[14] Sharmila, B. S., & Nagapadma, R. (2019). Intrusion Detection System using Naive Bayes algorithm. *2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*. https://doi.org/10.1109/wiecon-ece48653.2019.9019921

[15] Ayogu, B. A., Adetunmbi, A. O., & Ayogu, I. I. (2019). A comparative analysis of Decision Tree and Bayesian Model for Network Intrusion Detection System. *FUOYE Journal of Engineering and Technology*, *4*(2). https://doi.org/10.46792/fuoyejet.v4i2.362

[16] Amor, N. B., Benferhat, S., & Elouedi, Z. (2004). Naive Bayes vs decision trees in intrusion detection systems. *Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 14–17 March 2004*. https://doi.org/10.1145/967900.967989

[17] Panda M. and Patra, M.R. (2007) Network Intrusion Detection using Naive Bayes. Int. J. Comput. Sci. Netw. Secur. 7(12), 258–263.

[18] Yosifova V., A. Tasheva and R. Trifonov. (2021) Predicting Vulnerability Type in Common Vulnerabilities and Exposures (CVE) Database with Machine Learning Classifiers. *12th National Conference with International Participation (ELECTRONICA), Sofia, Bulgaria. pp. 1-6.* https://doi:10.1109/ELECTRONICA52725.2021.9513723

[19] Novianto, B., Suryanto, Y., & Ramli, K. (2021). Vulnerability Analysis of Internet Devices from Indonesia Based on Exposure Data in Shodan. *IOP Conference Series Materials Science and Engineering*, *1115*(1), 012045. https://doi.org/10.1088/1757-899x/1115/1/012045

[20] Chen Z. and Li Y. F., “Anomaly Detection Based on Enhanced DBScan Algorithm,” Procedia Engineering **15**, 178–182 (2011). doi: https://doi.org/10.1016/j.proeng.2011.08.036

[21] Sahoo, K., Samal, A. K., Pramanik, J., & Pani, S. K. (2019). Exploratory data analysis using python. International Journal of Innovative Technology and Exploring Engineering, 8(12), 4727–4735. https://doi.org/10.35940/ijitee.L3591.1081219