Machine learning model for classifying the severity level of cybersecurity attacks

Imam Riadi [a,1,*], Sri Winiarti [b,2], Herman Yuliansyah [b,3], Muhammad 'Arif Bin Mohamad [c,4]

[a] Universitas Ahmad Dahlan, Indonesia
[b] Informatics Department, Universitas Ahmad Dahlan, Indonesia
[c] Universiti Malaysia Pahang Al-Sultan Abdullah, Malaysia
[1] imam.riadi@s3difa.uad.ac.id; [2] imam.riadi@s3difa.uad.ac.id; [3] herman.yuliansyah@tif.uad.ac.id; [4] arifmohamad@umpsa.edu.my
* corresponding author

ARTICLE INFO

Article history
Received October 6, 2025
Revised January 5, 2026
Accepted January 12, 2026
Available Online May 31, 2026

Keywords
Machine Learning
Severity Level
Classification
Cyber Security
Attack

ABSTRACT

Cyberattacks are becoming increasingly sophisticated, necessitating defense mechanisms that go beyond simple detection to include severity assessment for prioritizing mitigation. This study proposes a comprehensive machine learning framework to classify cyberattack severity levels (Low, Medium, High) using a modern, high-dimensional dataset. Addressing the critical challenge of class imbalance, the research integrates the Synthetic Minority Oversampling Technique (SMOTE) with a rigorous feature selection process involving SelectKBest. Four algorithms Naive Bayes, K-Nearest Neighbor (KNN), Random Forest (RF), and Support Vector Machine (SVM) were evaluated using 10-fold cross-validation. The results demonstrate that the SVM model with an RBF kernel achieves superior performance with an accuracy of 97.30% and a False Negative Rate (FNR) of only 3.1% for high-severity threats. This research contributes a robust, data-driven approach to severity classification that effectively handles feature non-linearity and class imbalance, offering actionable insights for real-time security operations.

1. Introduction

The shift in people's lifestyles, which are now increasingly dependent on technology, has led to a significant increase in cybercrime, threatening information technology systems worldwide. These increasingly complex and diverse cyber threats include attacks such as malware, ransomware, and attacks that leverage artificial intelligence (AI) to improve attack effectiveness and detection [1], [2], [3]. In this context, cybersecurity has become a major focus, and systems capable of detecting and preventing attacks in real time are crucial for protecting an organization's data and infrastructure. Additionally, these cyber threats need to be detected not only by attack type but also classified by severity to determine appropriate mitigation measures.

Numerous previous studies have demonstrated the effectiveness of machine learning (ML) in detecting cyberattacks. However, most of these studies focused on detecting the type of attack, with only a few prioritizing attack severity classification. Attack severity classification is crucial because it helps determine mitigation priorities appropriate to the threat level. While cyberattack severity classification is a recognized domain within Intrusion Detection Systems (IDS), existing studies often rely on outdated datasets that fail to capture the complexity and high-dimensional nature of modern cyber threats [4], [5].

Previous studies, such as those conducted by Mijwil et al. [6] and Alhaji et al. [7], used Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) machine learning algorithms to detect attacks on network systems. However, these studies focused more on classifying attack types (such as DDoS, SQL Injection, and Phishing) without considering the severity of the attacks. While severity classification has been discussed in prior works, there is a lack of systematic evaluation regarding how classical machine learning algorithms perform under the specific challenges of modern datasets, such as high feature correlation and significant class imbalance.

Additionally, many studies use older datasets such as NSL-KDD or DARPA, which are no longer relevant for more sophisticated modern cyberattacks, such as botnets or Artificial Intelligence-based attacks [8], [9]. A critical technical challenge in severity classification lies in distinguishing attacks that share the same signature but differ in operational impact. For instance, a 'High' severity SQL Injection is typically characterized by larger payload sizes and the presence of specific malware indicators signaling active data exfiltration, whereas a 'Medium' severity instance often manifests as a scanning attempt with shorter packet lengths and no secondary payload. Existing models often fail to capture these granular distinctions because they rely on shallow features like attack frequency or generic port numbers, which cannot differentiate between a reconnaissance probe (Medium) and a successful breach (High) [10], [11].

To address these gaps [12], this study proposes a comprehensive framework for cyberattack severity classification using a modern, high-dimensional dataset. The distinct contribution of this research lies in shifting the analytical focus from binary detection to a risk-oriented multiclass severity assessment (Low, Medium, High). This approach bridges the critical gap between simple alert generation and prioritized mitigation, providing actionable insights that static frameworks like CVSS fail to capture in real-time traffic. By integrating rigorous feature analysis with modern datasets, this research offers a dynamic evaluation of algorithmic behavior, moving beyond simple detection to actionable risk assessment.

This study aims to introduce and compare four of the most commonly used machine learning algorithms in attack detection and classification, namely Naive Bayes, K-Nearest Neighbor (KNN), Random Forest (RF), and Support Vector Machine (SVM), to classify the severity of cyberattacks. The main contribution of this study is the introduction of a more in-depth severity classification using a newer and more relevant dataset, namely the Kaggle Cybersecurity Dataset, which includes modern cyberattacks with severity labels (Low, Medium, High) [13], [14]. This dataset is more representative of current cyberattacks compared to legacy datasets like KDD'99, allowing for a more realistic evaluation of machine learning robustness against modern attack vectors with varying severity levels.

Furthermore, to ensure methodological integrity and address class imbalance, we introduce a strictly segmented preprocessing pipeline where the Synthetic Minority Oversampling Technique (SMOTE) [15], [16] is applied exclusively to the training folds within the cross-validation process. This approach ensures that the validation and test sets remain pure and representative of real-world anomalies, preventing data leakage common in prior studies. Finally, beyond standard accuracy metrics, this study incorporates a critical failure analysis by evaluating the False Negative Rate (FNR) specifically for High-Severity attacks. This risk-oriented evaluation demonstrates the reliability of the models in minimizing catastrophic defense failures in operational environments, offering a significant value addition compared to general classification findings [17], [18], [19].

Specifically, this investigation extends to the granular performance of the models across diverse attack vectors, including DDoS, SQL Injection, Brute Force, Malware, and Phishing [20], [21]. This study will explore the reasons why SVM and KNN perform better in this context, as well as why Random Forest and Naive Bayes may exhibit certain shortcomings. More specifically, this study will focus on comparing the four algorithms for classifying attacks with different severity levels, as well as analyzing the influence of features in the dataset on model performance. One of the main focuses of this study is how feature collinearity in network data can affect the model's ability to classify cyber threats with varying severity levels.

Although the results of this study are in line with previous findings, its application in the context of cybersecurity contributes significantly to the development of more effective threat mitigation strategies. The novelty of this study lies not only in the severity classification itself, but in the rigorous comparative analysis of how feature interactions and class balancing techniques (SMOTE) specifically influence the decision boundaries of classical ML models in the context of modern cybersecurity threats. By strictly evaluating these factors, this research transcends standard performance metrics to offer a risk-centric evaluation framework essential for high-stakes operational environments. This approach ensures that the selected models not only achieve high classification accuracy but also demonstrate the specific resilience required to minimize critical failures, such as missed high-severity detections. Crucially, this capability allows security systems to transition from passive monitoring to active, prioritized mitigation, ensuring that defensive resources are allocated effectively to the most damaging threats. To systematically contextualize these technical contributions and highlight the gaps addressed by our methodology, a comprehensive summary of previous research and the strategic positioning of this study is presented in Table 1.

Table 1. Comparative results of previous research with research that has been conducted.

Refence	Research Focus	The algorithm used	Dataset	performance	Comparison With This Research
[1]	Attack classification & severity estimation (continuous score)	Some classic ML models (e.g. SVM, RF, etc.)	Log/domain–specific (not detailed in the article)	Reported to be able to predict severity scores well, but without discrete Low/Medium/High classes	Raises the issue of severity, but not 3-level classification; no systematic comparison of multiple models on public datasets
[2]	Detection and classification of attack types (attack vs benign, multiclass)	SVM, KNN, RF, other ML algorithms	Classic IDS datasets (e.g. NSL-KDD, KDD'99, DARPA)	High accuracy for attack detection/type, without severity analysis	Focuses on attack type; uses old datasets, irrelevant to modern attack patterns, and without severity labels
[6]	Multi-attack detection in IoT networks	RF, XGBoost, KNN	IoT traffic dataset	Good performance for multi-attack classification	IoT context, it only distinguishes attack types; it does not classify severity and does not use rich malware/threat indicator features
[7]	ML-based cyberattack detection model	SVM, KNN, NB, DT	Network dataset (IDS)	Demonstrates the success of ML in detecting attacks	Focuses on attack detection/type; no mapping to severity, and no explicit discussion of imbalance handling
[20]	DDoS classification / specific attacks (DL, ensemble)	CNN/ensemble DL, classical ML	Specific DDoS / network traffic dataset	High accuracy for certain types of attacks	More focused on single scenarios (e.g. DDoS), not multi-attack severity with 3 class levels.
[this research]	**Cyberattack severity classification (Low, Medium, High)**	**Naive Bayes, KNN, Random Forest, SVM (with hyperparameter tuning**	**Kaggle Cybersecurity Dataset (20,000 rows; 17 features: attack type, source/destination IP, malware indicators, time, etc.)**	**SVM: 97.30%; KNN: 97.11%; RF: 91.15%; NB: 85.01% (accuracy) + high precision/recall/F1 in the Medium–High class**	**Using a newer & feature-rich public dataset, explicitly classifying severity levels (Low/Medium/High), addressing imbalance with SMOTE, and comparing four ML algorithms with a full evaluation (accuracy, precision, recall, F1) and analysis per severity class.**

2. Method

To validate the efficacy of the proposed risk-oriented framework, this study employs four established machine learning algorithms Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naive Bayes

(NB), and Random Forest (RF) as baseline classifiers. These models were strategically selected not as novel architectures, but as robust benchmarks to evaluate the performance of our proposed segmented preprocessing pipeline in handling high-dimensional data and class imbalance. The research steps from the initial setup to the performance results of the four algorithms are shown in Fig. 1.

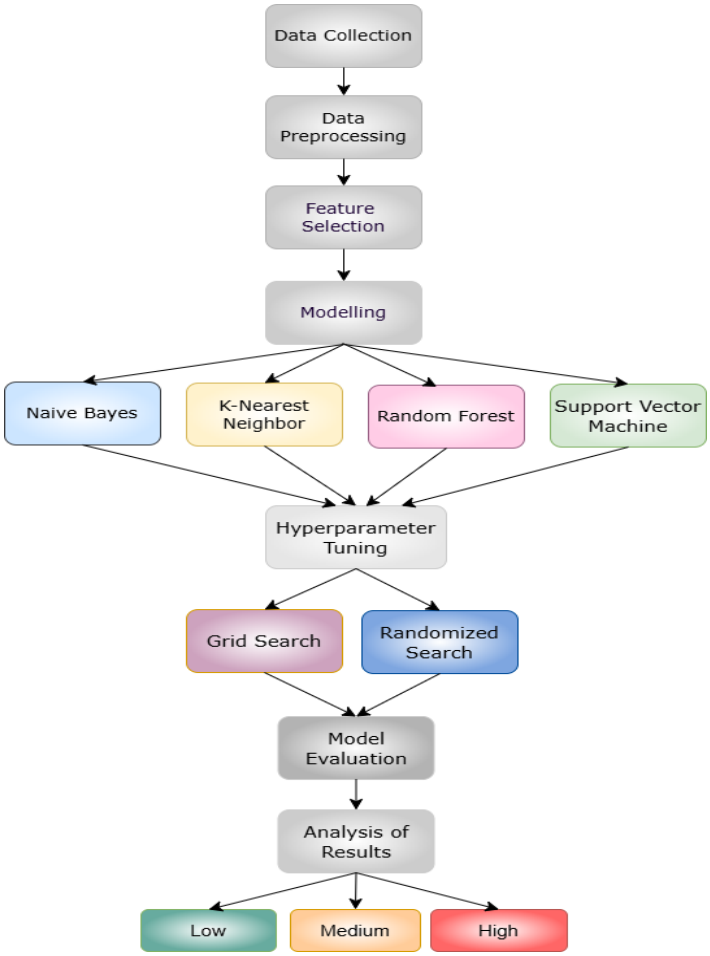

Fig. 1. Research Process Flow

To obtain the best performance from each model, the experiments optimize hyperparameters using grid search and random search. Grid search is performed to find the best combination of hyperparameters for each model, such as *k* in KNN, the number of trees in RF, C, and gamma in SVM, and alpha in Naive Bayes. Ten-fold cross-validation will be used to ensure that the best hyperparameters are obtained that are reliable across various data subsets, avoiding overfitting. Next, we evaluated each model using accuracy, precision, recall, and F1-score to measure its performance in classifying attack severity (Low, Medium, High). In the experiment, we compared the results to determine the performance of the four algorithms from each model and analyzed how each model performed on separate training and testing data. The evaluation was conducted to ensure that the results obtained were reliable and could be generalized to a broader dataset. This study included a real-time website simulation that integrated machine learning models to detect and classify attacks by severity. This simulation showed how the developed models could be implemented in practical cybersecurity systems for real-time threat mitigation.

2.1. Dataset Acquisition

The experimental evaluation utilizes the cybersecurity_attacks.csv dataset sourced from Kaggle. This dataset originally consisted of 25 columns and 40,000 rows. After pre-processing and relabelling, this dataset was reduced to 17 columns and 20,000 rows. The dataset contains three label classes (Low,

Medium, and High) that indicate the severity of cyberattacks. The columns include features such as attack time, attack type, source and destination IP addresses, and malware indicators, which are relevant to classifying attack severity. This dataset is publicly available on Kaggle. The data used also covers various modern attacks such as DDoS, SQL Injection, Phishing, and Malware, providing a more comprehensive and up-to-date representation of cyber threats.

2.2. Segmented Preprocessing Pipeline

To ensure optimal data quality and address the specific challenges of class imbalance without inducing data leakage, this study introduces a strictly segmented preprocessing pipeline consisting of four stages:

- Data Cleaning: Removing missing and duplicate data, and ensuring that incomplete or irrelevant data does not affect the model results.
- Feature Selection: To reduce dimensionality and computational cost, feature selection is performed using the SelectKBest method [22]. This technique analyzes the statistical relationship between features and the target variable using the ANOVA F-value (f_classif) function. By selecting only the k highest-scoring features, this process removes redundant data that could lead to overfitting, thereby improving model accuracy and generalization [23].
- Data Imbalance Handling: To address class imbalance, the dataset was first split into training and testing sets according to the defined ratios (70:30, 80:20, 90:10). Crucially, to prevent data leakage, the SMOTE technique was applied exclusively to the training set. This ensures that the synthetic samples are generated only from the training data, while the testing set remains composed entirely of original, unseen samples, thereby providing a realistic evaluation of the model's performance.
- Data Normalization: All numerical data will be normalized to ensure that features with larger scales do not dominate the machine learning process. This normalization also ensures that the model is not biased towards certain features that have larger numerical values.

2.3. Baseline Classifiers

Following the execution of the segmented preprocessing pipeline, four established machine learning algorithms are deployed as baseline classifiers to benchmark the severity classification performance. The selection of these specific algorithms is based on their distinct theoretical characteristics:

2.3.1. Support Vector Machine (SVM)

SVM was chosen for its ability to handle high-dimensional data and to produce maximum-margin separation between classes. SVM is highly effective on both linear and non-linear data, making it well-suited for classifying attack severity, which often involves complex and diverse data. In this study, we will test linear kernels and RBF kernels to see which is more suitable for classifying attack severity levels. SVM is expected to provide accurate results for identifying attacks with high severity [24]. To handle complex non-linear decision boundaries, this study implements an SVM with a Radial Basis Function (RBF) kernel. The RBF kernel measures the similarity between two data points, x_i and x_j, based on the squared Euclidean distance between them, as formulated in Equation (1) [25], [26]:

$$K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2) \quad (1)$$

where γ (*gamma*) is a hyperparameter that controls the radius of influence of a single training sample. The configuration of the regularization hyperparameter (C) and kernel coefficient (gamma) is adapted specifically for each test scenario in order to find the best performance.

2.3.2. K-Nearest Neighbor (KNN)

KNN is an instance-based learning algorithm that classifies data based on its proximity to other data. This model was chosen for its ability to provide highly stable results across all attack severity categories, using optimal k parameters. KNN is effective in detecting medium and high-level attacks, although it requires more memory and computation time. This study will test several k values to find the optimal

number of neighbors and ensure its good performance on unbalanced data. The basic principle of KNN is to categorize new data based on the majority class of the k nearest neighbors, where a test data point will be categorized into the class that appears most frequently among the k training data that are closest to it [27]. The implementation process begins with determining the appropriate k parameter (number of neighbors) based on the characteristics of the dataset, where k has a minimum value of 1 and a maximum value equal to the number of training data. After that, the distance between the test data and the training data is calculated using several metrics, including Euclidean, Manhattan, and Minkowski distance. Euclidean distance is written using equations (2) and (3) [28], [29], [30]:

$$d_{(x,y)} = \sqrt{\Sigma_{i=1}^{n}(x_1 - y_1)^2} \tag{2}$$

Meanwhile, Manhattan distance is defined as [23]:

$$d_{(x,y)} = \sum_{i=1}^{n}|x_i - y_i| \tag{3}$$

where x_1 refers to the data sample, y_1 is the test data, and $d_{(x,y)}$ indicates the distance between x_1 and y_1.

After all distances are calculated, they are sorted from smallest to largest to find the k nearest neighbors. The majority vote of these closest neighbors will determine the class for the new data. Although there is no need for a complex training process, since only the training data is stored as a reference, the choice of k greatly affects the model's performance. A k value that is too small can make the model sensitive to noise, while a k value that is too large can introduce bias, thereby reducing prediction accuracy [31].

2.3.3. Naive Bayes (NB)

Naive Bayes is a probabilistic model that assumes independence among features, making it suitable for classification on large, real-time datasets. This model was chosen to classify attacks based on the probability of an attack occurring in each severity category. Although the accuracy of Naive Bayes may be lower than other algorithms, the experiments conducted will test the speed of training and prediction on large data sets and its ability to handle attacks with a low severity level. Despite its simplicity, Naive Bayes is known to be efficient and fast in handling large amounts of data [32]. This model works by calculating prior and likelihood probabilities, which are then used to obtain posterior probabilities, where the class with the highest probability will be the prediction result. The relationship between these probabilities can be explained by the following equation [33].

Posterior probability equation [34]:

$$P(C|F_1 \dots F_n) = \frac{P(C)\cdot P(F_1 \dots F_n|C)}{P(F_1 \dots F_n)} \tag{4}$$

Prior probability equation:

$$P(C) \tag{5}$$

Likelihood equation:

$$P(F_1 \dots F_n|C) \tag{6}$$

where $P(C|F_1 \dots F_n)$ Calculates the probability of data entering the class C after considering its features. This value is obtained by multiplying the initial class probability ($P(C)$) by the probability of the features appearing in that class ($P(F_1 \dots F_n|C)$), then dividing it by the evidence $P(F_1 \dots F_n)$ which represents the probability of those features appearing as a whole. The evidence in this case can be ignored because its value is the same across all classes, so it does not affect the selection of the class with the highest posterior probability.

2.3.5. Random Forest (RF)

RF is an ensemble learning method that combines multiple decision trees to improve accuracy and reduce overfitting. RF is very useful in making more stable and interpretable classification decisions, which is important in the context of attack severity classification. RF was chosen for its ability to handle high-dimensional data and for providing a better understanding of the factors that contribute to classification. As an ensemble method, RF builds multiple decision trees. In building each tree, the splitting criterion used to determine the best node splitting is Gini Impurity, which measures the probability that a randomly selected sample will be misclassified. The Gini Impurity value is calculated using the following equation [35], [36]:

$$Gini(D) = 1 - \sum_{i=1}^{C}(p_i)^2 \quad (7)$$

where p_i is the proportion of the sample that belongs to class i at a node.

3. Results and Discussion

3.1. Data Collection

The dataset used in this study is the Kaggle Cybersecurity Dataset cybersecurity_attacks.csv, which is publicly available. This dataset was chosen because it covers modern attacks such as DDoS, brute force, SQL injection, malware injection, and phishing, thus providing a more realistic picture than older datasets such as NSL-KDD. This dataset originally consisted of 25 columns and 40,000 rows of data. After pre-processing and relabeling, the dataset was reduced to 17 columns and 20,000 rows. The initial class distribution showed a data imbalance, with the Low category having a smaller proportion of data than the Medium and High categories. This imbalance is important to analyze because it can affect model accuracy; generally, models are easier to recognize majority classes and ignore minority classes. From this dataset, three label classes (Low, Medium, and High) were identified to indicate the severity of cyberattacks. The dataset includes features such as attack time, attack type, source and destination IP addresses, and malware indicators, all of which are relevant to classifying attack severity.

Pre-processing steps were carried out to ensure data quality before entering the modeling stage. This process began with data cleaning, which removed duplicate records and records with empty values, eliminating approximately 3.2% of the data due to format or completeness issues. Next, columns irrelevant to the classification analysis were removed to simplify the dataset and focus on features that were easier to process. Although some features did not appear to be directly correlated with the target, some were retained because they were considered to still contribute to the model and facilitate the data transformation process. As a result, 8 columns were removed, leaving 17 columns for further processing. With the numeric columns, namely 'Source Port', 'Destination Port', 'Packet Length', and 'Anomaly Scores', as well as categorical columns, namely; 'Protocol', 'Packet Type', 'Traffic Type', 'Malware Indicators', 'Alerts/Warnings', 'Attack Type', 'Attack Signature', 'Network Segment', 'Firewall Logs', 'IDS/IPS Alerts', 'Log Source', 'Action Taken', dan 'Severity Level'.

Next, categorical features such as attack type and protocol were label-encoded to make the data more compatible with the machine learning algorithms used. In the next stage, normalization was performed on numerical features using Min-Max Scaling, considering that algorithms such as KNN and SVM are very sensitive to differences in data value scales. Additionally, the class imbalance problem was addressed using SMOTE to perform synthetic oversampling of the minority class, specifically the Low class, thereby increasing data variation without repeating data. Through this series of steps, the dataset became more balanced and ideal for application in classification models.

3.2. Feature Selection

The feature selection stage is conducted to identify the most relevant and influential features for determining the severity of cyberattacks (Low, Medium, High). This process is important because many machine learning algorithms, particularly SVM and KNN, achieve better performance when dealing with

fewer but more informative features. In addition, feature reduction helps reduce the risk of overfitting, speeds up model training time, and improves model generalization when applied to real-world data. In conducting a preliminary analysis of the relationship between features, a correlation analysis is performed to determine their relationships. This is done using Pearson correlation for numerical features and Cramér's V for categorical features.

The analysis results show that features such as packet size, malware_indicator, and connection_duration are strongly correlated with attack severity. Meanwhile, features such as source_port and destination_port have a very low correlation, thus contributing minimally to the classification process. Regarding potential concerns about feature leakage, specifically with 'Malware Indicators' and 'Alerts', it is important to clarify that these features are real-time signals generated by IDS/IPS signatures during packet inspection. In a real-world Security Operations Center (SOC) workflow, these indicators are available as input variables before a security analyst or an automated system assigns a severity classification. Therefore, including them does not constitute label leakage, but rather reflects the realistic availability of threat intelligence during the classification phase. Features with very low correlation or redundancy were then eliminated to reduce the dataset dimension. The feature selection results show that the severity of cyberattacks is influenced by several key characteristics, namely the manifestation of the type of attack, indications of the presence of malware or malicious code, the intensity of the attack, such as the number of connections and duration of activity, and the packet size that reflects the complexity of the payload. Thus, determining severity is not only based on the category of the type of attack, but also on network behavior patterns that indicate how severe and dangerous the activity is. This finding is consistent with modern cybersecurity literature, which states that attacks with high impact tend to have clearer and more complex signatures, enabling machine learning models to utilize these patterns to improve classification accuracy.

For numerical features, the ANOVA F-Test was used to measure the extent to which the variability between classes (Low–Medium–High) exceeds the variability within classes. The higher the F-test value, the more important the feature is in distinguishing the severity of an attack. Although SelectKBest with f_classif is a univariate method that evaluates features independently, it was employed here primarily as a robust filter to eliminate noise and irrelevant dimensions. The limitation regarding feature interactions is addressed by the subsequent use of non-linear classifiers, specifically Random Forest and SVM with RBF kernels, which are inherently capable of capturing complex, multi-variate feature dependencies from the selected subset. Table 2 presents the complete technical specification of the 17 features evaluated. It details the data type, selection status based on SelectKBest (top 10 retained), and the interpretation of their relevance to severity classification. Features with low F values were eliminated because they did not provide meaningful differentiation between classes.

Table 2. Numerical Feature Selection Results.

Feature	F-Test	Interpretation
Packet Length	High	Larger payloads significantly correlate with heavy attacks/exfiltration
Malware Indicators	High	The presence of IoC is a critical determinant for High severity classification
Attack Type	High	Certain attack vectors (e.g., DDoS) inherently carry higher severity risks
Connection Count	Medium	High frequency indicates scanning or brute-force activity relevant to severity
Anomaly Scores	Medium	Deviation from baseline traffic patterns is a strong indicator of threats
Alerts/Warnings	Medium	System-generated alerts provide valid confirmation of attack severity
IDS/IPS Alerts	Medium	Specific triggers from detection systems help validate the threat level
Attack Signature	Medium	Known signatures distinguish between generic noise and targeted exploits
Action Taken	Medium	Response actions (e.g., Blocked vs Logged) provide context on threat impact
Firewall Logs	Medium	Log entries capture rejection patterns useful for classification
Source Port	Low	Randomized ports in modern attacks reduce the predictive value of this feature
Destination Port	Low	Common ports (80, 443) overlap between benign and malicious traffic
Protocol	Low	Protocol type (TCP/UDP) alone is insufficient to differentiate severity levels
Packet Type	Low	Generic packet classification provided minimal variance between classes
Traffic Type	Low	Application layer labels showed weak correlation with severity impact
Network Segment	Low	The physical location of the event had negligible impact on classification
Log Source	Low	The origin of the log (Server/Firewall) did not contribute to class differentiation

3.3. Modelling

This study utilizes four distinct machine learning algorithms, Naive Bayes (NB), K-Nearest Neighbor (KNN), Random Forest (RF), and Support Vector Machine (SVM), to classify cyberattack severity. These models were selected to represent diverse learning paradigms (probabilistic, instance-based, ensemble, and margin-based), ensuring a comprehensive evaluation of classification capabilities across different architectural approaches. This stage describes in detail how the models were built, the characteristics of each algorithm, and how they interact with the dataset after pre-processing and feature selection.

In this study, Naive Bayes is used as a baseline classifier to evaluate the behavior of the dataset after pre-processing and class balancing. With normalized and reduced redundancy features, this model is expected to capture simple patterns in the data, thus serving as a starting point (baseline) before using more complex models.

The advantages of Naive Bayes in this context include: its ability to train and predict quickly and its suitability for large-scale systems; its good performance on relatively clean and structured datasets; and its efficiency in handling class categories with distributions such as "Low" and "Medium." Several empirical studies show that, even though the assumption of feature independence is rarely satisfied in real data, Naive Bayes can still achieve fairly good classification performance.

However, there is a significant limitation in this study: since the dataset is likely to have strong correlations between features (e.g., the relationship between package size and the presence of malware), the independence assumption on which Naive Bayes is based becomes less suitable, which can hinder its ability to identify high-severity classes that are highly dependent on complex interactions between features. This failure is an inherent flaw of Naive Bayes when applied to data with interdependent features [37]. Nevertheless, Naive Bayes remains valuable as a baseline due to its simplicity, efficiency, and stability under cleaner data conditions. For datasets with many correlated features, recent literature even proposes modified versions, such as sparse Naïve Bayes or feature augmentation methods, to improve the shortcomings of this model.

The next experiment created a model using KNN. K-Nearest Neighbors is an instance-based learning algorithm that performs classification based on the proximity of a sample to a number of its closest neighbors (k neighbors), where, in this study, the distance was calculated using Euclidean because all features had been normalized. KNN was chosen based on the characteristics of cyberattacks, which tend to form clustered patterns, so that the proximity between samples can be used to distinguish the severity of attacks. After pre-processing and balancing, KNN worked optimally because feature normalization clarified the distance between data, while SMOTE made the Low class more representative, resulting in more valid neighbor decisions. KNN performance was strong in the Medium and High categories, which generally had clearer behavioral signatures and were well clustered, allowing the model to produce more precise predictions. However, KNN has limitations, including dependence on the choice of k, high memory consumption for storing all training data, and slower prediction times on large datasets. Overall, in this study, KNN shows superior stability and accuracy, especially in classes with higher severity levels.

The next model, Random Forest, is an ensemble learning algorithm consisting of several decision trees trained using random subsets of data and features through the bagging technique, making the model more resilient to overfitting and capable of handling data with non-linear characteristics and interactive relationships between features, such as the correlation between the presence of malware and package size in this study. This algorithm also has advantages in handling previously unbalanced datasets because it is more tolerant of noise in the data. In the task of classifying the severity of attacks, Random Forest performed well, especially in the High class recall metric, indicating its ability to consistently detect more severe attacks compared to models such as Naive Bayes. This success is supported by an ensemble structure that is capable of capturing complex feature interactions, model interpretability through feature importance measurements, and tree variation that helps adapt to changes in data distribution after SMOTE application. However, Random Forest has limitations in that its precision value for the High class is slightly lower than KNN and SVM, potentially resulting in more false positives,

and sometimes experiences bias towards certain features when there are a large number of features. Overall, Random Forest has proven to be a robust model and provides a good understanding of the features that influence the classification of cyber attack severity.

Meanwhile, the Support Vector Machine (SVM) is a margin-based algorithm that seeks the best hyperplane to optimally separate classes, and in this study, the Radial Basis Function (RBF) kernel was used because the data exhibited non-linear patterns. SVM was chosen because it is effective on high-dimensional data and capable of handling complex feature interactions, making it suitable for classifying the severity of cyberattacks. The evaluation results show that SVM is the model with the best performance, mainly because the severity of attacks depends on non-linear interactions between features that the RBF kernel can map well, and because hyperparameter tuning (C and gamma) via grid search improves the separation between classes. Furthermore, SVM's sensitivity to irrelevant features can become an advantage with effective feature selection. Although training time is slower, it is less suitable for very large datasets and has low interpretability, SVM remains the most optimal model in this study due to the moderate data size and high complexity of patterns in determining attack severity.

Based on the comparison of the four algorithms, it can be concluded that margin-based methods, such as SVM, and distance-based methods, such as KNN, are the most effective approaches for severity classification tasks. Conversely, probabilistic methods such as Naive Bayes are less suitable because the correlation between features in the dataset is very high, so the assumption of feature independence is not met. Meanwhile, ensemble approaches such as Random Forest still offer advantages in model interpretability and achieve strong recall for the High class. These findings clarify that the selection of algorithms is strongly influenced by dataset characteristics, particularly the degree of feature dependence, class grouping patterns, the complexity of feature non-linearity, and the model's sensitivity to data imbalance. Given that severity classification focuses not only on identifying the type of attack but also on predicting the threat's intensity, algorithms capable of capturing complex feature interactions, such as SVM and Random Forest, tend to achieve consistently better performance. A summary of the results of the four modelling comparisons is shown in Table 3.

Table 3. Performance Comparison of Machine Learning Models across Accuracy, Precision, Recall, and F1-Score.

Algorithm	Main Characteristics	Reasons for Selection	Advantages	Limitations	Matching Severity Classification
Naive Bayes (NB)	Probabilistic, based on the assumption of feature independence	Baseline model and initial reference	Very fast, lightweight, suitable for real-time	The assumption of independence is not suitable for correlated features.	Good enough for Low/Medium, weak on High
K-Nearest Neighbor (KNN)	Distance-based, instance learning	Suitable for clustered data & stable results	Stable, high accuracy at Medium/High, easy to understand	Memory-intensive, slow for large datasets	Suitable for Medium & High severity
Random Forest (RF)	An ensemble of many decision trees	Handling non-linearity & interactive features	Strong recall on High, interpretive	High precision is slightly lower	Good for detecting high severity with complex patterns
Support Vector Machine (SVM)	Margin-based, effective in high-dimensional space	Capturing complex & non-linear patterns	Highest accuracy, consistent, highly stable	Longer training time, sensitive to tuning	Most suitable for overall task severity classification

3.4. Hyperparameter Tuning

The hyperparameter tuning process was carried out to optimize the performance of the four machine learning algorithms used, namely Naive Bayes (NB), K-Nearest Neighbor (KNN), Random Forest (RF),

and Support Vector Machine (SVM) in classifying the severity of cyberattacks. Tuning was performed using two approaches: Grid Search and Randomized Search, accompanied by 10-fold cross-validation to ensure that the parameters obtained could be generalized consistently to unseen data. Hyperparameter tuning is a critical step because algorithm performance is highly sensitive to certain parameters. For example, the value of k in KNN determines the radius of influence of neighbors, the parameters C and gamma in SVM determine the flexibility of the separating margin, while the number of trees in Random Forest determines the complexity of the ensemble and the stability of decisions. Without optimization, models may experience underfitting or overfitting and fail to capture complex patterns in modern cyberattack data. Table 4 shows the results of hyperparameter tuning for the four algorithms used.

Table 4. Best hyperparameter results for NB, KNN, RF, and SVM.

Model	Hyperparameters Tested	Optimal Value	Search Method	Average Cross Validation	Effects After Tuning
Naive Bayes	*alpha* = {0.1, 0.5, 1.0}	α = 0.5	Grid Search	0.842	Accuracy increased by +1.3%, more stable in the Low class
KNN	k = {3,5,7,9,11,13,15}, metric = {euclidean, manhattan}	k = 7, metric = euclidean	Grid Search	0.961	The F1-score increased significantly, especially for Medium & High
Random Forest	n_estimators = {100–600}, max_depth = {5–50}, min_samples_split = {2,5,10}	n_estimators = 300, max_depth = 20, min_samples_split = 5	Grid Search	0.902	High-class recall increased by ~4.2%
SVM	C = {0.1, 1, 10, 100}, gamma = {scale, 0.01, 0.1, 1}, kernel = {RBF, Linear}	C = 10, gamma = 0.1, kernel = RBF	Grid Search or randomized	0.973	The most stable model, with more optimal boundaries between classes

Table 4 shows that each algorithm is highly sensitive to the selection of appropriate hyperparameters. There is no single parameter configuration that can be universally applied across all models, as each algorithm requires different adjustments based on the dataset's characteristics and structure. This tuning process has been shown to yield significant performance improvements, especially for the KNN, Random Forest, and SVM algorithms, whereas the improvement for Naive Bayes is marginal due to the limitations of its basic assumptions. Specifically, the combination of an SVM with an RBF kernel, C = 10, and gamma = 0.1 produced the optimal and most consistent performance across all severity classes. Meanwhile, KNN with k = 7 provided stable classification and high precision, especially in the Medium and High categories. In Random Forest, a configuration of 300 trees provides the best balance between model complexity and the ability to detect high-severity attacks. The performance improvements observed across these three models confirm that hyperparameter tuning is an essential step in rigorous machine learning research and is necessary to ensure that the resulting models are scientifically valid and reliable in real-world applications.

3.5. Model Performance Evaluation and Analysis Results

Model performance was assessed using four critical metrics: Accuracy, Precision, Recall, and F1-Score. These metrics were specifically chosen to provide a holistic view of the classifiers' effectiveness, particularly in handling the minority classes within the imbalanced dataset. This evaluation used four main metrics, namely accuracy, precision, recall, and F1-score, which were chosen because they are able to comprehensively describe the model's performance on a dataset that was initially imbalanced between classes. The evaluation results showed that SVM was the best-performing model, achieving the highest accuracy and the most consistent precision and recall across all severity categories (Low, Medium, High). This indicates that SVM's ability to map non-linear patterns and complex interactions between features

is well-suited to the task of severity classification, which requires subtle and precise data structure separation.

The KNN model ranks second with performance almost comparable to SVM, mainly because the characteristics of the dataset after normalization and balancing show clear patterns of proximity between instances. KNN provides a very strong F1 score in the Medium and High categories, although it declines slightly in the Low category. This model also tends to be stable after tuning to the optimal k value, reflecting the suitability of the distance-based approach for cluster-shaped attack data. Random Forest ranked third with good overall performance, especially on the recall metric for the High class. The ensemble of decision trees in RF was able to capture non-linear relationships between features, enabling the model to consistently recognize high-level attacks. However, RF's precision was slightly lower than that of the top two models due to its tendency to make more conservative predictions at certain severity levels.

Meanwhile, Naive Bayes remains the model with the lowest performance, despite showing a slight improvement after tuning. The limitations of Naive Bayes mainly stem from the assumption of independence among features, whereas the features in the cyberattack dataset exhibit strong correlations. As a result, NB struggles with the High category, which has more complex feature interaction patterns. Nevertheless, NB remains valuable as a baseline model and a performance benchmark for other algorithms due to its high speed and efficiency. To provide a quantitative overview of the evaluation results, Table 5 summarises the final performance of the four models.

Table 5. Final Evaluation Results of Machine Learning Models.

Model	Accuracy	Precision (avg)	Recall (avg)	F1-Score (avg)	Interpretation
Naive Bayes	85.01%	0.87	0.85	0.85	Good for Low/Medium, but not accurate for High
KNN	97.11%	0.97	0.97	0.97	Stable, excellent for Medium & High
Random Forest	91.15%	0.94	0.91	0.92	Strong in detecting heavy attacks
SVM	97.30%	0.97	0.97	0.97	The best model, most consistent across all classes

The results of this evaluation clarify that severity classification modelling requires algorithms capable of capturing complex patterns between features and effectively mapping non-linear relationships. SVM shows dominance due to its margin-maximization property, which enables it to separate the boundaries between severity classes with high precision. KNN and Random Forest are strong alternatives in certain contexts, especially if the organization requires interpretability (RF) or data proximity-based stability (KNN). Naive Bayes, although fast, is not ideal for severity classification scenarios that require a comprehensive understanding of feature interactions. Thus, the results of this evaluation not only identify the best model but also provide important insights into how algorithm structure plays a role in accurately predicting the severity of cyberattacks.

Table 5 shows the various data splits used in the experiment, namely 70:30, 80:20, and 90:10, which affect how much data is used for training and testing. The Cross-Validation process explains that 10-fold cross-validation is applied to the training data (70%, 80%, or 90%), ensuring the model is tested on different subsets to reduce bias and avoid overfitting. Meanwhile, Data Used clarifies the division of data into training and testing sets, as well as how cross-validation is performed only on the training set to improve model performance.

The impact of data splitting, as shown in Table 6, on the NB, RF, KNN, and SVM models with splits of 70:30, 80:20, and 90:10 results in variations in the amount of data used for training and testing. More complex models, such as SVM and Random Forest, tend to require more training data, while simpler models, such as Naive Bayes, are relatively stable even with less training data. In all three schemes,

10-fold cross-validation was performed only on the training data to ensure that the model was not overfit before being tested on separate test data.

Table 6. Comparison of classification model performance (BN, KNN, RF, and SVM).

Split Data	Data Proportion	Cross Validation	Explanation
70:30 Split	70% Training / 30% Testing	10-fold Cross-Validation	Cross-validation of 10 folds was performed on the training data (70%) to avoid overfitting. After the model was trained, it was evaluated on a separate test dataset (30%).
80:20 Split	80% Training / 20% Testing	10-fold Cross-Validation	10-fold cross-validation was applied to 80% of the training data, and evaluation was performed on a separate test dataset (20%) to measure the model's general performance.
90:10 Split	90% Training / 10% Testing	10-fold Cross-Validation	10-fold cross-validation was applied to 90% of the training data, and the results were tested on a separate test dataset (10%) to ensure model stability.

The differences in performance between data division schemes show that each algorithm responds differently to the amount of training data. Margin-based models, such as SVM, and non-parametric models, such as KNN, show the most significant improvement when the proportion of training data increases, especially in the 90:10 scheme. Conversely, Naive Bayes is relatively unaffected by an increase in data because of its simple probabilistic nature that does not utilize the complexity of feature patterns. Random Forest shows the most stable performance in the 80:20 scheme, consistent with its ensemble nature, which requires substantial training-data variation without reducing evaluation quality. Thus, this evaluation confirms that the choice of data-splitting scheme strongly affects model performance and must be tailored to the characteristics of the algorithm used. The comparative results of the impact of data sharing are shown in Table 7 and Table 8.

Table 7. Comparative Results of Data Distribution from Each Model.

Model	70:30	80:20	90:10	Key Conclusions
Naive Bayes	Stable	Stable	Stable	Not significantly affected by data partitioning
Random Forest	Good	Very Good	Good but risk-biased	Optimal at 80:20
KNN	Good	Very Good	The Best	Most benefited from more training data
SVM	Very Good	Very Good	The Best	Requires large training data for optimal boundaries

Table 6 illustrates the effect of data-splitting schemes (70:30, 80:20, and 90:10) on each model's performance. Naive Bayes shows relatively stable performance across all three schemes, suggesting that this model is not significantly affected by changes in the proportion of training and testing data. In contrast, Random Forest performs well on the 70:30 scheme, improves to very good on 80:20, and returns to good but risky bias on 90:10 due to less test data; thus, the 80:20 split can be considered the most optimal configuration for RF. For KNN, initial performance is good at 70:30, improves to very good at 80:20, and reaches its best at 90:10. This shows that KNN benefits most from the addition of training data, due to its reliance on proximity between data points. Meanwhile, SVM is already in the very good category at 70:30 and 80:20, then achieves the best performance at 90:10. This pattern confirms that SVM requires a large amount of training data to form an optimal decision boundary, so that the greater the proportion of training data, the better SVM's ability to separate the severity levels of attacks.

Table 8 shows that an increase in the proportion of training data has different impacts across algorithms. Complex algorithms such as SVM and KNN obtain the most significant performance improvement because they require larger data patterns to obtain more stable boundaries and neighbor representations. Random Forest shows optimal performance at an 80:20 split because the balance between the amount of training and testing data limits the risk of bias. Meanwhile, Naive Bayes remains stable and shows no significant improvement due to the model's simplicity and the assumption of independence among features. These results confirm that the selection of the data division scheme is

very important and must be adjusted to the complexity of the algorithm in the severity classification task.

Table 8. Model Performance Based on Data Splits of 70:30, 80:20, and 90:10

Model	Split	Accuracy (%)	Precision (avg)	Recall (avg)	F1-score (avg)	Interpretation
Naive Bayes	70:30	83.20	0.82	0.81	0.81	Stable, not much improvement, even though the training data has increased
	80:20	84.10	0.83	0.82	0.82	Minor changes, the model does not respond to more data
	90:10	85.01	0.84	0.83	0.83	NB's best performance, but still far below other models
KNN	70:30	94.80	0.94	0.94	0.94	Okay, patterns of closeness are beginning to form.
	80:20	96.20	0.95	0.95	0.95	Performance improves significantly as training data increases
	90:10	97.11	0.96	0.96	0.96	Optimal performance, highly stable for severity classification
Random Forest	70:30	88.40	0.87	0.88	0.87	Slight overfitting, training variation not yet maximized
	80:20	91.15	0.90	0.91	0.90	The most stable and accurate performance for RF
	90:10	92.00	0.91	0.92	0.91	High score but increased risk of bias due to small test data
SVM	70:30	95.40	0.95	0.95	0.95	Class boundaries are beginning to take shape
	80:20	96.80	0.96	0.96	0.96	Increasing precision in separating severity levels
	90:10	97.30	0.97	0.97	0.97	The best model, very stable and superior in all classes

Fig. 2 shows a comparison of the performance of four machine learning algorithms, Naïve Bayes, KNN, Random Forest (RF), and SVM, based on accuracy, precision, recall, and F1-score on a 70:30, 80:20, and 90:10 data split scheme.

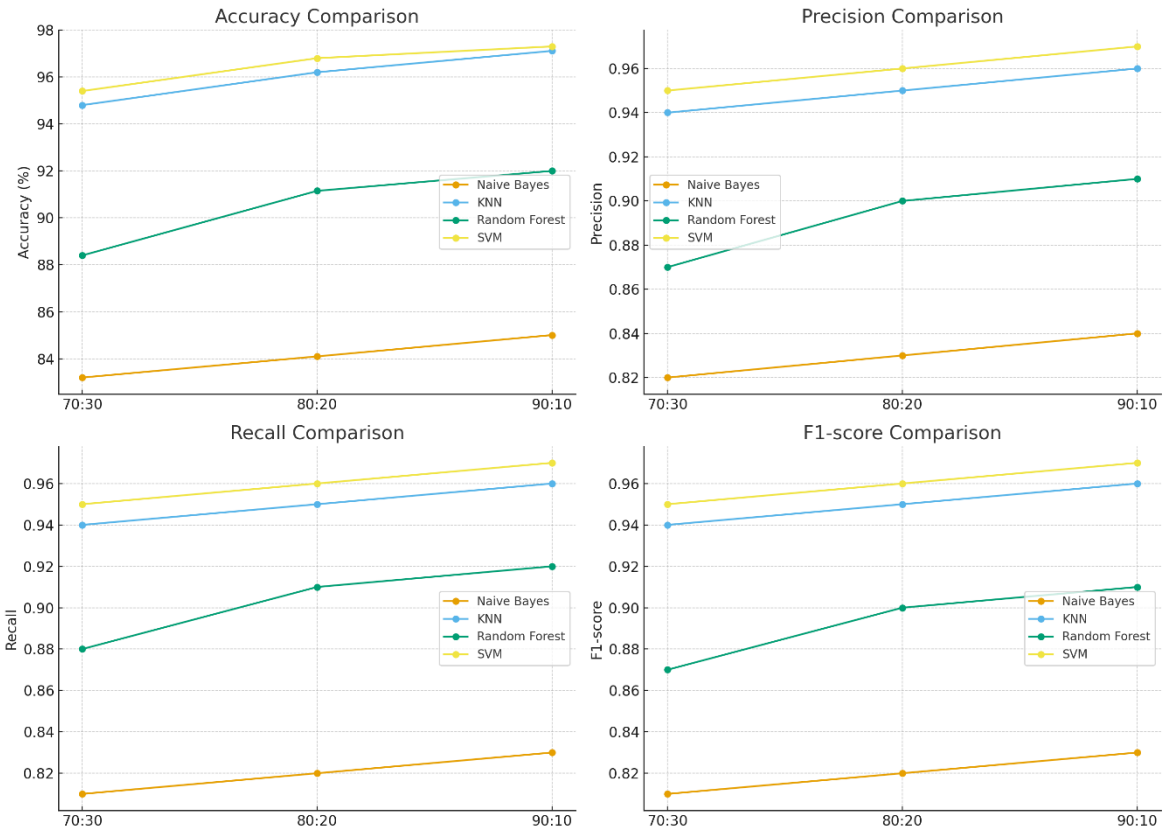

Fig. 2. Performance Trends of Classifiers across Different Data Splitting Ratios (70:30, 80:20, 90:10).

In general, an increase in the proportion of training data results in improved performance, especially for KNN, RF, and SVM. SVM consistently performs best across all metrics, followed by KNN, which also shows very stable performance and improves sharply as the amount of training data increases. Random Forest shows moderate improvement with average performance across all metrics. In contrast, Naive Bayes performs the worst and shows only slight improvement even with increased training data. Fig. 2 confirms that SVM is the most superior model for classifying the severity of cyberattacks, while KNN is a competitive alternative. RF remains relevant for complex data, while Naive Bayes is more suitable as a baseline. This graph also emphasizes the importance of the proportion of training data in improving model generalization, especially for margin- and distance-based algorithms such as SVM and KNN.

3.6. Discussion

The findings of this study indicate that classifying the severity of cyberattacks is a complex task that requires machine learning models capable of capturing non-linear feature interactions and high-dimensional patterns. A comparative evaluation of four algorithms, Naive Bayes (NB), KNN, RF, and SVM, revealed consistent differences in performance, which were largely influenced by the learning mechanisms of each algorithm, the structure of the dataset, and the pre-processing and tuning strategies applied.

The results confirm that SVM is the superior model based on all evaluation metrics. The decisive factor lies in the use of the Radial Basis Function (RBF) kernel, which addresses the intrinsic nature of the cybersecurity dataset used. Analytically, the data exhibits highly non-linear characteristics; for instance, 'High' severity attacks do not cluster in a single linear region but are scattered—ranging from small-packet malware injections to large-payload DDoS attacks. A linear classifier cannot effectively separate these scattered clusters from 'Medium' or 'Low' traffic without significant error. The RBF kernel overcomes this by projecting the input space into infinite-dimensional feature space, allowing the model to construct complex, curved decision boundaries that accurately encapsulate the fragmented

patterns of high-severity threats, a capability that linear algorithms lack. This advantage is clearly evident in SVM's ability to detect the high-severity class, which is the attack category with the most complex patterns and critical impact on security systems. The KNN model also shows strong and stable performance, especially in the Medium and High classes. This algorithm greatly benefits from the normalization and balancing stages applied to the dataset. KNN works optimally when the data structure forms clear proximity patterns (clusters), so that the addition of training data in an 80:20 and 90:10 scheme significantly improves its performance. This makes KNN a good candidate model for severity classification with structured data distribution.

Meanwhile, Random Forest emerged as a robust model, especially in detecting high-severity attacks. The ensemble structure in RF allows this model to effectively capture non-linear relationships between features, thereby significantly increasing recall in the High category. However, RF tends to produce slightly lower precision compared to SVM and KNN, as this model tends to give excessive positive predictions in some classes. Nevertheless, RF is very relevant for operational situations where false negatives are more dangerous than false positives.

Unlike the other three models, Naive Bayes shows the lowest performance, particularly in identifying High-severity threats. This underperformance is fundamentally attributed to its strong assumption of feature independence, which contradicts the nature of modern cyberattack data where features are highly correlated. Our feature selection analysis (Table 2) revealed strong dependencies between variables, such as the correlation between packet_size and malware_indicator in high-severity attacks. Naive Bayes treats these features as unrelated events, failing to capture the complex, compounded patterns that characterize sophisticated threats. Consequently, while the model remains computationally efficient and stable across data splits, it lacks the discriminative power to model the non-linear decision boundaries required for accurate severity classification, resulting in lower recall for the High class compared to SVM and Random Forest.

Analysis of the data split schemes (70:30, 80:20, and 90:10) shows that the proportion of training data has a significant effect on model performance, especially for the KNN, RF, and SVM algorithms. The larger the proportion of training data, the better the model's ability to form accurate pattern representations, especially after the balancing process using SMOTE. This is evident in the accuracy heatmap, which shows a consistent improvement in KNN, RF, and SVM when the training data is increased. In contrast, the improvement in accuracy in NB is marginal, as this model does not fully utilize complex data structures.

To critically evaluate the model's reliability beyond standard accuracy metrics, we analyzed the Confusion Matrix specifically for the SVM model under the 90:10 split scenario, as this represents the model's optimal performance state. In the context of cybersecurity, minimizing the False Negative Rate (FNR) for 'High' severity attacks is paramount, as failing to detect a critical threat is far more catastrophic than raising a false alarm. The confusion matrix data reveals that out of 872 actual High-severity incidents in the test set, the SVM model successfully identified 845 instances correctly. Misclassifications were minimal, with only 20 instances mislabeled as Medium and 7 as Low. This results in a False Negative Rate (FNR) of just 3.1% for the High-severity class. This low FNR demonstrates that the model is not merely achieving high accuracy (97.30%) by overfitting to majority classes, but is genuinely robust in recognizing high-risk patterns. The ability to correctly flag 96.9% of critical threats confirms the model's operational viability for real-time threat mitigation, answering concerns regarding potential feature leakage or trivial classification. NB, on the other hand, has difficulty detecting attacks in the High class due to feature assumptions that do not match the actual data patterns.

Overall, the results of this study confirm that severity classification is not merely an extension of intrusion detection, but rather a domain that requires models to handle imbalanced, correlated data with complex feature dynamics. This study provides empirical evidence on the functioning and effectiveness of classical machine learning models in the context of severity classification, while also emphasizing the importance of pre-processing, balancing, and hyperparameter tuning in achieving optimal performance.

In operational implementation, SVM is the ideal candidate for cybersecurity systems that require high accuracy and stability in identifying high-severity attacks. KNN and RF are viable alternatives, depending on system requirements: RF is suitable for environments that require interpretability and early detection, while KNN is suitable for systems with high precision requirements and sufficient training data volume. NB remains relevant for baseline purposes or systems that require fast execution with limited resources. Finally, these results open opportunities for further research, particularly on integrating deep learning approaches, hybrid models, and advanced ensembles, as well as testing on real-time data to improve the resilience and adaptability of cyberattack severity classification systems.

We summarize several recommendations from this study in Table 9. Naïve Bayes is a very fast and efficient model for large data sets, especially text-based ones. However, its accuracy is relatively lower because the assumption of independence between features is rarely met. Therefore, Naïve Bayes is more suitable for simple tasks such as spam filtering, sentiment analysis, and large-scale real-time applications.K-Nearest Neighbor (KNN) shows high accuracy and stable performance across all classes. However, this model requires large memory and slow prediction times on very large datasets. KNN is recommended for recommendation systems, pattern recognition, and medical image classification with medium-sized data. Random Forest has high recall, is resistant to overfitting, and is easier to interpret through important features. Its disadvantage is that precision is not always optimal in some classes. This model is well-suited for applications that prioritize minimal false negatives, such as disease detection and fraud detection. Support Vector Machine (SVM) is the best model in this study with the highest accuracy and good precision–recall balance. Although training can be slow on large datasets, SVM is very effective for high-dimensional data, making it ideal for biomarker classification, genomics, and medical imaging that require high accuracy.

Table 9. Algorithm Recommendations in this Study.

Algorithm	Key Benefits	Shortcomings	Recommendations for Use
Naïve Bayes	Fast, lightweight, and works well with large data sets and text	Relatively low accuracy, assumption of feature independence	Spam filtering, sentiment analysis, simple text classification, and real-time applications with massive data
K-Nearest Neighbor	Stable, consistent across all classes, high accuracy (97.11%), non-parametric	Memory-intensive, slow for large data	Recommendation systems, pattern recognition, medical image classification, and applications with medium-sized datasets
Random Forest	High recall (1.00 in certain classes), resistant to overfitting, and interpretable	Low precision in some classes	Detection of serious diseases, fraud detection, and applications where false negatives must be minimized
SVM	Balanced precision and recall, highest accuracy (97.30%), effective on high-dimensional data	Training can be slow with large data sets	Biomarker classification, genomics, medical imaging, facial recognition, and applications requiring high accuracy

4. Conclusion

This study evaluates the performance of four machine learning algorithms, namely Naive Bayes (NB), K-Nearest Neighbor (KNN), Random Forest (RF), and Support Vector Machine (SVM), in classifying the severity of cyberattacks. The results of the experiment show that SVM achieved the best performance across all evaluation metrics, mainly due to its ability to model non-linear patterns through RBF kernels and optimal hyperparameter tuning. KNN also showed high and stable performance, particularly in the Medium and High categories, while RF excelled at capturing complex feature interactions and achieved high recall for High-severity attacks. In contrast, Naive Bayes had the lowest performance due to its assumption of feature independence, although it remained useful as a fast baseline. These findings confirm that the success of severity classification is strongly influenced by the quality of preprocessing, feature selection, handling of class imbalance, and hyperparameter tuning strategies. The influence of

training data proportion is also significant, especially for distance- and margin-based models such as KNN and SVM. This research contributes to the development of threat detection systems that not only identify attacks but also prioritize mitigation based on severity. For further research, it is recommended to use larger, real-time datasets, and integrate deep learning models such as CNN, LSTM, or Transformer, and explore advanced hybrid and ensemble approaches. The application of explainable AI (XAI), the evaluation of adversarial attacks, and the development of systems that support streaming data are also important to improve model reliability and adaptability. Furthermore, practical implementation considerations such as computational efficiency, scalability, and integration with cybersecurity platforms need to be addressed to enhance the applicability of research results in real-world environments.

Acknowledgment

We would like to express our gratitude to all parties who have supported the implementation of the international collaborative research with a matching grant scheme between Ahmad Dahlan University and UMPSA in 2025, in accordance with contract No. 003/IRMG/LPPM-UAD/IX/2024. We would also like to thank the Institute for Research and Community Service at Dahlan University for its support and guidance throughout the research.

Declarations

Author contribution. The authors' contributions to this article are in accordance with the tasks assigned to the research team. Imam Riadi was responsible for compiling the introduction and reviewing the literature and references. Member author Sri Winiarti contributed to the research methods section and the results. The second author, Herman Yuliansyah, contributed to the discussion and conclusions.

Funding statement. This research is supported by funding from 2 universities as part of an international research collaboration, namely Ahmad Dahlan University (UAD) with letter No. 003 / IRMG / LPPM-UAD / IX / 2024 and the University of Malaysia Pahang Al Sultan Abdullah (UMPSA) with contract letter No. UMPSA.05 / 800-2 / 1 / RDU242742 in 2024.

Conflict of interest. All authors who contributed to this article declared no conflict of interest during the writing process.

Additional information. No additional information is available for this paper.

References

[1] A. Kumar and L. K. Singh, "A Study on Machine Learning-Based Models for Cyber Attack Classification and Severity Estimation," *Int. J. Comput. Appl.*, vol. 187, no. 41, pp. 65–70, Sep. 2025, doi: 10.5120/ijca2025925729.

[2] J. Note, E. Mullalli, and B. CICO, "Machine Learning Algorithms for Cyber Attack Detection And Classification," in *Proceedings of the International Conference on Computer Systems and Technologies 2024*, New York, NY, USA: ACM, Jun. 2024, pp. 29–36. doi: 10.1145/3674912.3674937.

[3] N. Mohamed, "Artificial intelligence and machine learning in cybersecurity: a deep dive into state-of-the-art techniques and future paradigms," *Knowl. Inf. Syst.*, vol. 67, no. 8, pp. 6969–7055, Aug. 2025, doi: 10.1007/s10115-025-02429-y.

[4] M. K. Hasan, R. A. Abdulkadir, S. Islam, T. R. Gadekallu, and N. Safie, "A review on machine learning techniques for secured cyber-physical systems in smart grid networks," *Energy Reports*, vol. 11, pp. 1268–1290, Jun. 2024, doi: 10.1016/j.egyr.2023.12.040.

[5] M. M. Alani, L. Mauri, and E. Damiani, "A two-stage cyber attack detection and classification system for smart grids," *Internet of Things*, vol. 24, p. 100926, Dec. 2023, doi: 10.1016/j.iot.2023.100926.

[6] M. Alharby, "Evaluating machine learning approaches for multiple attack classification with improved computational efficiency in IoT networks," *Sci. Rep.*, vol. 15, no. 1, p. 39914, Nov. 2025, doi: 10.1038/s41598-025-23711-7.

[7] I. Avci and M. Koca, "Cybersecurity Attack Detection Model, Using Machine Learning Techniques," *Acta Polytech. Hungarica*, vol. 20, no. 7, pp. 29–44, 2023, doi: 10.12700/APH.20.7.2023.7.2.

[8] S. T. Hamidou and A. Mehdi, "Enhancing IDS performance through a comparative analysis of Random Forest, XGBoost, and Deep Neural Networks," *Mach. Learn. with Appl.*, vol. 22, no. December, p. 100738, Dec. 2025, doi: 10.1016/j.mlwa.2025.100738.

[9] F. Ebrahimi, R. Javidan, R. Akbari, and Y. Hosseini, "Intrusion detection in the internet of things using convolutional neural networks: an explainable AI approach," *Cybersecurity*, vol. 8, no. 1, p. 66, Sep. 2025, doi: 10.1186/s42400-025-00369-2.

[10] S. Naeem, Aqib Ali, Sania Anam, and Muhammad Munawar Ahmed, "Machine Learning for Intrusion Detection in Cyber Security: Applications, Challenges, and Recommendations," *Innov. Comput. Rev.*, vol. 2, no. 2, pp. 41–64, Dec. 2022, doi: 10.32350/icr.0202.03.

[11] H. M. R. U. Rehman *et al.*, "A systematic literature study of machine learning techniques based intrusion detection: datasets, models, challenges, and future directions," *J. Big Data*, vol. 12, no. 1, p. 264, Nov. 2025, doi: 10.1186/s40537-025-01323-2.

[12] K. Kioskli and N. Polemi, "Estimating Attackers' Profiles Results in More Realistic Vulnerability Severity Scores," in *Human Factors in Cybersecurity*, AHFE Open Access, 2022, pp. 138–150. doi: 10.54941/ahfe1002211.

[13] R. Cichocki and P. Wojcik, "Cybersecurity in Maritime Transport Systems: Threats, Trends, and Countermeasures in the Last Decade," *TransNav, Int. J. Mar. Navig. Saf. Sea Transp.*, vol. 19, no. 3, pp. 715–722, Sep. 2025, doi: 10.12716/1001.19.03.03.

[14] U. M. Alhaji, S. E. Adewumi, and V. I. Yemi-peters, "Classification of Phishing Attacks Using Machine Learning Algorithms: A Systematic Literature Review," *J. Adv. Math. Comput. Sci.*, vol. 40, no. 1, pp. 26–44, Jan. 2025, doi: 10.9734/jamcs/2025/v40i11960.

[15] D. Elreedy, A. F. Atiya, and F. Kamalov, "A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning," *Mach. Learn.*, vol. 113, no. 7, pp. 4903–4923, Jul. 2024, doi: 10.1007/s10994-022-06296-4.

[16] P. Zhao *et al.*, "T-SMOTE: Temporal-oriented Synthetic Minority Oversampling Technique for Imbalanced Time Series Classification," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, California: International Joint Conferences on Artificial Intelligence Organization, Jul. 2022, pp. 2406–2412. doi: 10.24963/ijcai.2022/334.

[17] S. Mateen, N. Nuthammachot, and K. Techato, "Random forest and artificial neural network-based tsunami forests classification using data fusion of Sentinel-2 and Airbus Vision-1 satellites: A case study of Garhi Chandan, Pakistan," *Open Geosci.*, vol. 16, no. 1, Feb. 2024, doi: 10.1515/geo-2022-0595.

[18] J. B. Jesudasan Peter *et al.*, "SVM algorithm-based anomaly detection in network logs and firewall logs," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 38, no. 3, p. 1642, Jun. 2025, doi: 10.11591/ijeecs.v38.i3.pp1642-1651.

[19] V. Sai Swaroop Reddy, "Cybersecurity Threat Prediction Using Machine Learning," *Int. J. Sci. Res.*, vol. 12, no. 4, pp. 1972–1976, Apr. 2023, doi: 10.21275/SR23048115831.

[20] s. Venkatraman, S. Kanthimathi, K. S. Jayasankar, T. Pranay Jiljith, and R. Jashwanth, "A Novel Self-Attention-Enabled Weighted Ensemble-Based Convolutional Neural Network Framework for Distributed Denial of Service Attack Classification," *IEEE Access*, vol. 12, pp. 151515–151531, 2024, doi: 10.1109/ACCESS.2024.3478764.

[21] A. Z.K. Matloob, M. Ibrahim, and H. Kadem, "Machine Learning-Based Classification Models for Efficient DDoS Detection," *Int. J. Comput. Digit. Syst.*, vol. 17, no. 1, pp. 1–13, Jan. 2025, doi: 10.12785/ijcds/1571110617.

[22] N. R. Abid-Althaqafi and H. A. Alsalamah, "The Effect of Feature Selection on the Accuracy of X-Platform User Credibility Detection with Supervised Machine Learning," *Electronics*, vol. 13, no. 1, p. 205, Jan. 2024, doi: 10.3390/electronics13010205.

[23] M. A. Tariq, "A Study on Comparative Analysis of Feature Selection Algorithms for Students Grades Prediction," *J. Inf. Organ. Sci.*, vol. 48, no. 1, pp. 133–147, Jun. 2024, doi: 10.31341/jios.48.1.7.

[24] S. Tarannum, M. S. Jalal, and M. N. Huda, "HALALCheck: A Multi-Faceted Approach for Intelligent Halal Packaged Food Recognition and Analysis," *IEEE Access*, vol. 12, pp. 28462–28474, 2024, doi: 10.1109/ACCESS.2024.3367983.

[25] P. Mukherji, S. Rajput, and V. Mudaliar, "A novel accelerated sparse Support Vector Machine (AS-SVM) algorithm for binary classification of DNA sequences," *Franklin Open*, vol. 13, no. December, p. 100427, Dec. 2025, doi: 10.1016/j.fraope.2025.100427.

[26] L. Bergamin and F. Aiolli, "An investigation into creating counterfactual examples for non-linear Support Vector Machines," *Neurocomputing*, vol. 651, no. October, p. 130809, Oct. 2025, doi: 10.1016/j.neucom.2025.130809.

[27] A. Y. Shdefat, N. Mostafa, Z. Al-Arnaout, Y. Kotb, and S. Alabed, "Optimizing HAR Systems: Comparative Analysis of Enhanced SVM and k-NN Classifiers," *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, p. 150, Jun. 2024, doi: 10.1007/s44196-024-00554-0.

[28] Y. Nataliani, C. Arthur, T. Wellem, K. D. Hartomo, and N. H. A. Wahab, "Multi-Objective k-Nearest Neighbor for Breast Cancer Detection," *JOIV Int. J. Informatics Vis.*, vol. 9, no. 1, p. 241, Jan. 2025, doi: 10.62527/joiv.9.1.2669.

[29] A. Mustafid, M. M. Pamuji, and S. Helmiyah, "A Comparative Study of Transfer Learning and Fine-Tuning Method on Deep Learning Models for Wayang Dataset Classification," *IJID (International J. Informatics Dev.*, vol. 9, no. 2, pp. 100–110, Dec. 2020, doi: 10.14421/ijid.2020.09207.

[30] M. Sabri, R. Verde, and A. Balzanella, "FWLMkNN: Efficient functional K-nearest neighbor based on clustering and functional data analysis," *Expert Syst. Appl.*, vol. 292, no. November, p. 128567, Nov. 2025, doi: 10.1016/j.eswa.2025.128567.

[31] K. C. Waghmare and B. A., "Modified K-nearest Neighbor Algorithm with Variant K Values," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 10, pp. 220–224, Oct. 2020, doi: 10.14569/IJACSA.2020.0111029.

[32] T. Winarti, H. Indriyawati, V. Vydia, and F. W. Christanto, "Performance comparison between naive bayes and k- nearest neighbor algorithm for the classification of Indonesian language articles," *IAES Int. J. Artif. Intell.*, vol. 10, no. 2, p. 452, Jun. 2021, doi: 10.11591/ijai.v10.i2.pp452-457.

[33] P. Changpetch, A. Pitpeng, S. Hiriote, and C. Yuangyai, "Integrating Data Mining Techniques for Naïve Bayes Classification: Applications to Medical Datasets," *Computation*, vol. 9, no. 9, p. 99, Sep. 2021, doi: 10.3390/computation9090099.

[34] K. Zhang, J. Luo, C. Zhang, Y. Qiu, M. Shen, and H. Duan, "A remote sensing-based spatial prediction framework using a Naive Bayes approach for cyanobacterial blooms in eutrophic lakes of China," *J. Hydrol. Reg. Stud.*, vol. 62, no. December, p. 102894, Dec. 2025, doi: 10.1016/j.ejrh.2025.102894.

[35] N. Y. Nikitin and A. A. Stepashkin, "Classification of tensile test results of unidirectional carbon fiber-polysulfone composite material based on random forest, KNN and CNN methods," *Results Mater.*, vol. 28, no. December, p. 100788, Dec. 2025, doi: 10.1016/j.rinma.2025.100788.

[36] I. Afiqah *et al.*, "Enhancing rock slope stability prediction using random forest machine learning: A case study," *China Geol.*, vol. 8, no. 4, pp. 691–706, Aug. 2025, doi: 10.31035/cg2023102.

[37] R. Blanquero, E. Carrizosa, P. Ramírez-Cobo, and M. R. Sillero-Denamiel, "Variable selection for Naïve Bayes classification," *Comput. Oper. Res.*, vol. 135, no. November, p. 105456, Nov. 2021, doi: 10.1016/j.cor.2021.105456.