Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 11, 2025

CYBERSECURITY CHALLENGES AND THE ROLE OF MACHINE LEARNING IN MODERN MALWARE DETECTION

Muhammad Irfan Akram* [1], Shazia Yousaf[2], Muhammad Zubair[3], Malaika Riaz [4], Muhammad Sarmad shakil[5], Muhammad Yousif[6]

[1,5]School of computer science, Minhaj University, Lahore, Pakistan.
[2] Lecturer Department of Computer Science, Fazaia College of Education for Woman, Affiliated with Air University, Lahore, Pakistan
[3]Department of Computer Science, Air University Islamabad, Multan Campus, Multan 60000, Pakistan
[4]School of computer science, Khwaja Fareed University of Engineering & Information Technology, Rahim Yar Khan, Pakistan
[6]Department of Computer Science, National University of Modern Languages, Campus-Lahore, Pakistan
irfanakram.cs@mul.edu.pk*[1], shaz.fcoe@gmail.com[2], mzubair.ciit020@gmail.com[3], malaikariaz3210@gmail.com[4], sarmadshakil34@gmail.com[5], muhammad.yousif@numl.edu.pk[6]

DOI: https://doi.org/

Keywords
real-time systems, learned data structures, reinforcement learning, concurrent indexes, tail latency, energy-proportional computing, algorithmic accountability

Article History
Received on 20 September 2025
Accepted on 02 October 2025
Published on 27 October 2025

Corresponding Author: *
Muhammad Irfan Akram

Abstract

The cybersecurity has already become one of the most significant concerns of the digital world of the global community as the number of online users is increasing exponentially, and the organizations become more interconnected because of the interconnection of applications and services. Cyberattacks have reached levels of high risks with the increased use of online systems by business and government to communicate, transact financial transactions and share information. These threats exploit the network vulnerabilities, software vulnerabilities and human vulnerabilities leading to monetary loss, information breach and reputation loss. Viruses, worms, Trojan horses, DDoS attacks, phishing, and ransomware are among the activities that have become more advanced and popular as well as a trend in hackings and other activities that constitute frauds. Hackers have also become better in circumventing the old security systems by developing advanced evasion techniques, automation and artificial intelligence. Such threats do not just annihilate sensitive information data but also mess up vital infrastructures like health care, banking and government services. Cybersecurity is thus not a luxury that can be invested on but a necessity of every organization that upholds integrity, availability and confidentiality of data. Organizations use a mix of technical and procedural defenses as a way of mitigating these risks. Firewalls, network arrangements, and Intrusion Detection Systems (IDS) are some of the tools that are necessary in the monitoring, detection and prevention of illegal access. Firewalls are used as controls between trusted and untrusted networks whereas the network traffic is analyzed by the IDS to detect suspicious activities. Nevertheless, an increasing sophistication of modern malware is a significant challenge. A large number of the malicious

programs is script-based, polymorphic, or embedded in legitimate files and can therefore not be detected by the traditional antivirus systems or operating systems. In addition, attack patterns are constantly being changed by cybercriminals which requires security systems to keep evolving. The attack surface is now bigger than ever, with the emergence of Internet of Things (IoT) devices and cloud computing. The technologies have connected billions of devices all over the world thereby providing more points of attack to attackers. Therefore, advanced threat detection systems have become relevant. The recent few years have been characterized by the emergence of the notion of machine learning (ML) and artificial intelligence (AI) as the promising tools that can be used to identify and label malicious activity with higher accuracy. Some of the popular algorithms used in the domain of ML to detect the malware patterns and any anomalies in large volumes of data include Random Forest, Naive Bayes, K-Nearest Neighbor (KNN), Support Vector machine (SVM) and Logistic Regression. The algorithms will enhance cybersecurity since they will learn automatically on the history data, detect new attack signatures, and respond to new threats. The performance metrics with respect to determining the effectiveness include precision, accuracy, recall, and the Receiver Operating Characteristic (ROC) curve. The symbiosis of the ML-based systems with the traditional cybersecurity framework will be a more proactive, flexible, and smart approach to the digital infrastructure security since cyber threats are constantly evolving.

1. INTRODUCTION

The internet connectivity, cloud computing, and Internet of Things (IoT) have erupted with blistering development, which has seen the variety and quantity of cyber threats reach novel levels. Of the threats, malware malicious programs which target to interfere, destroy, or misuse computer systems has continued to be one of the most dynamic and enduring threats. In fact, cybercriminals are devising new forms of malware which are excavating loopholes within the networks and applications and even the hardware components. Consequently, ensuring both the securities and the information of such attacks would require the implementation of the traditional security practices alongside the superior, flexible and intelligent security practices. One such most rewarding such is machine learning (ML), that can be used to secure computers to identify as well as act on attacks through cyber threats [1] with minimal human contro.

Machine learning algorithms have proved extraordinary success in a range of applications, from natural image recognition to stock market forecasts, and their use in cybersecurity specifically malware detection has caused particular interest. The core advantage of ML-based malware detection is that it can be trained on past data, detect complex behaviors, and generalize from known to unknown threats. Differently from traditional signature-based systems that use pre-defined rules or known malware prints, the ML algorithms can be trained over enormous corpuses of benign as well as malicious files to detect unseen hidden associations as well as behavioral traits. This enables the systems based on ML to recognize the zero-day attacks, the polymorphic malware, as well as different sophisticated threats that go-un-noticed-on-traditional-Avs[2].

Nevertheless, as the potential of machine learning in cybersecurity is great, its execution is full of technical, moral, as well as working challenges. They emanate both from the nature of malware as well as the limitations of existing ML methods. For example, machine learning systems need big, varied, as well as rightly labeled datasets to be highly precise. However, in practice, such datasets are hard to get because of data privacy issues, imbalance, as well as obfuscation. The malware creators are known to utilize code obfuscation, packing, as well as encryption to hide the nature of the true behaviors of malicious files [3], thus making it hard to differentiate between the harmful as well as the benign activities by the use of the systems based on ML. Moreover, the dynamic nature of the malware variant requires that the models always be re-trained as well as re-updated with high computational cost as well as complexity of-maintenance.

One specific significant challenge is the adversarial nature of cybersecurity. Unlike most other uses of ML, malware detection occurs in the context of attacks that explicitly try to fool detection models. Adversary machine learning allows attackers to subtly corrupt features or implant misleading samples in the training data such that the system incorrectly classifies malware as harmless software. Attacks based on poisoning and evasion that occur in this manner pose the vulnerability of current ML models as an important problem of their faithfulness and stability. The resulting game of cat and mouse between both the creators of detection algorithms

and the creators of evasions enables a continuous evolutionary change that makes static or hard-coded ML models uncompetitive over the long term.

Another significant concern is model interpretability and transparency. Most of the most efficient machine learning methods, including deep learning and ensemble models, are "black boxes" they generate precise predictions but do not provide any explanation of how the predictions are made. In cybersecurity applications, it is unfruitful to lack explainability because analysts and organizations must know why a specific file or action behaved in a malicious way. Without interpretability, it becomes hard to corroborate the results of detection, debug false positives, or be in compliance with and auditing purposes.

Additionally, where the decision to secure automates, the inability to explain the behavior of the model can undermine the confidence of the people as well as impede the usage of the tools based on the ML in the systems that are crucial. False positives as well as false negatives pose significant practical challenges. A false positive in which benign software incorrectly matches to malicious malware can cause unnecessary disruptions, loss of reputation, and loss of funds. A false negative in which malicious code is missed can allow significant breaches as well as exfiltration of data. Blending this trade-off between specificity as well as sensitivity is one of the most precarious tasks in crafting ML-based malware detectors [4] Too sensitive models may overwhelm the analysts with alarms, whilst too permissive models may fail to detect real threats. To achieve this accurately requires feature building accuracy, threshold adjustment and even repeated model evaluation. Moreover, malware is dynamic and heterogeneous, which complicates feature extraction, which is the foundation of good machine learning. Malware may be analyzed by use of either of the two features; the static features (e.g. binary signature, or sequence of opcodes or imported functions) or dynamic features (e.g. system calls, network activity or API traces during execution) [5]. The two present trade-offs. The former is faster and safer, and it can be fooled by obfuscation, whilst the latter can provide much more insight into behavior, but is computationally expensive and more difficult to scale. Combination of hybrid models that combine the two strategies is becoming a trend that could address the problem, but it also introduces another burden in data integration, normalization, and real-time implementation.

Another issue is the computational and resource requirements of ML-based systems, which could be viewed through the prism of infrastructure [6]. Deep learning models need large processing units, memory, and storage in order to be trained on millions of malware samples. It might not be feasible to install such systems at scale in small and medium-sized organizations without cloud-based or distributed architectures. Nonetheless, the use of cloud computing is associated with a set of security threats such as exposing of data and compliance particularly where sensitive threat intelligence data is required to be shared or stored by other groups.

There are also ethical [7] and regulatory consequences of using machine learning in cybersecurity. The data that is used to train malware detection models may include sensitive user information, proprietary code or samples collected on the compromised systems. The issue

of privacy and data integrity and compliance with legal regulations, such as the General Data Protection Regulation (GDPR)[8], is a significant challenge to maintain. Along with that, automated systems making decisions about the possible cyber threats should be made responsible and equitable. There is a threat of discriminative training information whereby a certain software or programmers are incorrectly recognized, which raises the question of discrimination and threat category being employed without its rightful use.

Nevertheless, the possible advantages of machine learning in malware detection are almost impossible to underestimate because of these challenges. ML models can greatly decrease the labor input of people when correctly implemented and automated to perform the threat classification, prioritize the alerts, and detect the new attack patterns with a higher speed than they would be detected by a human analyst. More sophisticated methods like deep neural networks, reinforcement learning and graph-based models are under investigation in order to increase accuracy and flexibility. As an example, graph neural networks can be used to determine the connections among functions, system calls, or network entities to identify coordinated malicious behavior. Correspondingly, ongoing learning algorithms may assist models to change to new malware families without retraining afresh, enhancing long-term sustainability.

In order to proceed, cybersecurity scientists and professionals should pay attention to the creation of expedient, decipherable, and adaptive ML models. This will involve the integration of adversarial defense systems, enhance the diversity of datasets, and explain models that can provide an understanding of how they arrive at their decisions. Academia, industry, and government collaboration can help, as well, in the development of standardized datasets, open benchmarks, and common threat intelligence platforms. This kind of cooperation is needed to hasten innovation without compromising security as well as ethical standards.

To sum up, machine learning is not a silver bullet even though it can be a transformative power in the malware war. These obstacles related to data quality, adversarial manipulation, interpretability, scalability, and ethical issues suggest the difficulty of adapting ML to the reality of cybersecurity systems. The malware detection future is not only tied to advances in technology in terms of algorithms but also to the overall capacity of researchers and defenders to counter an ever-changing strategy of cyber attackers [9]. Finally, to create a secure and resilient digital ecosystem, it will need a consistent stream of innovations, monitoring, and collaboration between machine learning and cybersecurity.

1. Literature Review

Machine learning (ML) has come to be a core of the new malware detection, including the promise of generalization beyond signatures and responsiveness to the rapidly changing threat: but current literature highlights ongoing issues with dataset quality, adversarial robustness, explainability, deployment issues, and lifecycle security risks[10]. As noted by recent reviews, a lack of data, the presence of classes, and obfuscation are all fundamental barriers to sound generalization. According to Bensaoud et al. (2024), the inability to compare the performance of different DL strategies across Windows,

Android, iOS, macOS, and Linux fairly due to the absence of standardized and fresh benchmarks in particular in the case of the zero-day families is an issue. Molina-Coronado et al. (2023) indicate that concept drift severely affects batch detectors in the case of Android, yet retraining strategies based on drift detection and selective sampling might restore the performance effectively [11]. In 2025, more empirical evidence will further measure the impact of drift on various feature regimes, such as static, dynamic, hybrid, semantic, and image-based, and show that performance will decline over time as families change [12]. The surveys across platforms concur that these issues of datasets/features and transferability are multiplied by the heterogeneity between PCs, mobile, IoT, and cloud [7].

The systematic research on valid adversarial examples of binaries and the relative strength of malware generators is one of those trends since 2024. Kozak et al. (2024) present reinforcement-based learning-induced perturbations preserving functionality and being reliably able to evade detectors in realistic (black-box) settings [13]. Louthanova et al. (2024) also give a comparison of adversarial malware generators of Windows PE, which can reliably pass through several detection pipelines, which also means that generators are becoming more developed and can be applied to various systems [14]. The 2025 umbrella survey of adversarial attacks on deep networks brings taxonomies and trends together across domains, and this statement reinforces the assertion that diversity in attacks and realism in evaluation is not performing well in most studies [15]. In line with these investigations, the 2025 NIST taxonomy solidifies the ambitions of attackers, their knowledge, and a life-cycle, and brings together standardized terms to model threats to the security-critical environment of ML systems [16]. Combined together, these contributions render anti-evasion strength, poisoning hardiness, and assessment benchmarks very imperative research goals of malware ML.

Cited	Dataset Used	Model / Approach	Limitations	Advantages / Contributions
[17]	Android Drebin & CICMalDroid	Concept drift handling with retraining and sampling	Limited to Android; requires frequent updates	Demonstrated efficient drift adaptation using selective retraining
[18]	Mixed-platform malware corpora	Survey of ML algorithms for multi-platform malware	Absence of standardized cross-platform benchmarks	Unified analysis of ML across PC, mobile, IoT, and cloud
[19]	Custom enterprise malware dataset	Ensemble ML (Random Forest + SVM hybrid)	High computational cost, latency in real-time	Improved detection accuracy and robustness over traditional AV systems
[20]	IoT-23	Lightweight ML	May	Energy-efficient,

		(KNN, DT, RF)	underperform vs DL in complex malware	real-time IoT deployment feasible
[21]	Windows & Linux malware CFGs	Hierarchical Attention Graph Neural Network (MalHAPGNN)	High training time, complex tuning	Superior accuracy & feature interpretability through attention mechanisms

2. Proposed Methodology

As shown in figure 1, a full machine learning (ML) workflow of detecting malware can be subdivided into two primary steps, namely Training and Validation. Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression algorithms are some examples of algorithms that can be used in the ML Training Phase to utilize labeled Training Data to construct an ML model that learns to distinguish between Benign and Malware samples. After the training of the model, it enters the ML Validation Phase whereby it is tested on other Validation Data to determine its accuracy and reliability. These are examined by analyzing the results based on a decision flow: in case the model is able to classify new samples properly, it means that the model performs well, whereas misclassification indicates a bad performance, and thus requires retraining or changes in the parameters. This step is a feedback mechanism of improving the model to achieve greater accuracy of malware detection.

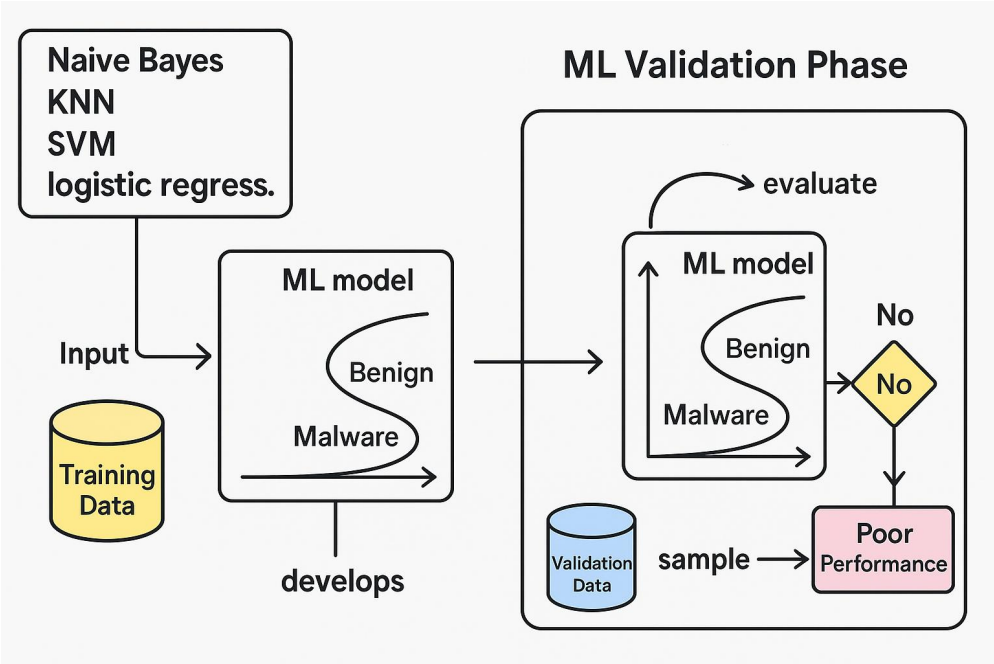

Figure 1. Proposed Methodology for Malware Detection

3.1 Data Collection and Preprocessing

Relevant Data Acquisition: It will entail the search and acquisition of cybersecurity data to train and test the model. Data Cleaning and Transformation: Preprocessing to clean up the missing information and outliers in order to have quality data.

3.2 Model Selection and Evaluation

Choice of Algorithms: Choosing the most appropriate machine learning algorithms i.e. Naive Bayes, support vector machines, k-nearest neighbor, or logistic regression depending on the nature of the problem and the data. Training and Testing: Splitting of the dataset into training and testing, application of appropriate sample size, and comparison of the model generalization. Performance Metrics: In order to define performance of the models, it will be required to find performance metrics including accuracy, precision, recall, F1-score, as well as area under the curve (AUC).

3.3 Deployment and Integration

Real time Monitoring: Implementation of models into live systems which would facilitate monitoring of cyber threats in real time and real time response to the threats. Integration with Security Infrastructure This represents a scenario where the learned model based on machine learning is integrated with an existing security system, e.g., firewall or intrusion detection system. Model Updates and Maintenance: There is a need to implement systems that will refresh models with new data and keep pace with the evolving threat environment.

3. Results & Simulation

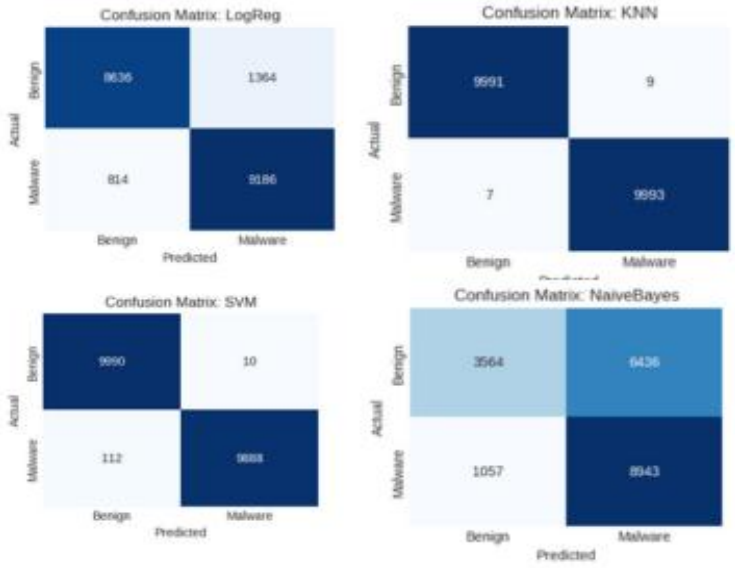

Figure 2. Confusion Matrix for All Models

Figure 2 summarize the confusion matrices demonstrate that, although with varying degrees of accuracy, all four models performed well in differentiating between benign and malicious samples. KNN and SVM demonstrated exceptional dependability in detecting malware, achieving the greatest accuracy and misclassifying only a few cases in each class. Although it generated more false positives and false negatives than KNN and SVM, logistic regression nevertheless fared well. With a significantly higher frequency of misclassifications, particularly for innocuous samples incorrectly classified as malware, Naïve Bayes demonstrated the worst performance. Overall, KNN and SVM showed better consistency and discrimination abilities across both classes, making them the most successful classifiers in this dataset.

Based on model performance for evaluating different metrices are discussed:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

$$\text{F1 Score} = 2*\frac{Precision*Recall}{Precision+Recall} \quad (3)$$

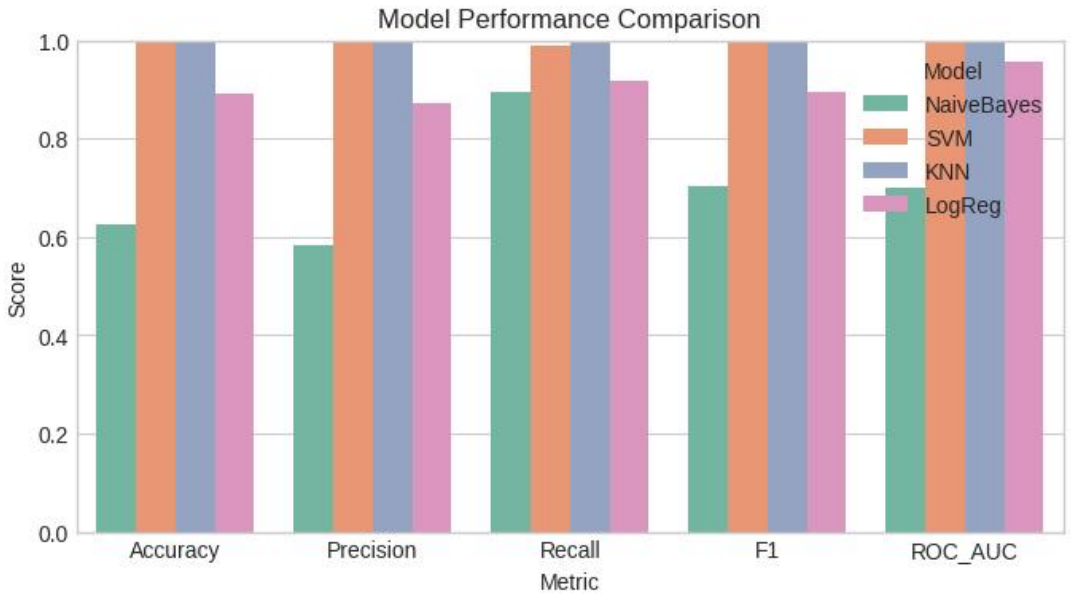

Figure 3. Histogram Plot for Comparison

Figure 3 display the model performance comparison chart demonstrates that each algorithm differs compared to the others in the way in which they tackle the classification task by a variety of evaluation metrics. Both SVM and KNN are always in the top or close to the top in terms of accuracy, precision, recall, F1 and ROC-AUC, which means they are highly reliable and balanced predictor. The results of the Logistic Regression are also consistent, although they are slightly lower than SVM and KNN, the score on all measures is high. Conversely, Naive Bayes performs poorly in front of the other models, particularly in the areas of precision and F1, but its recall is relatively better. In general, the plot indicates that SVM and KNN are the best models to use in this data, whereas Logistic Regression is a good compromise, and Naive Bayes is less predictive.

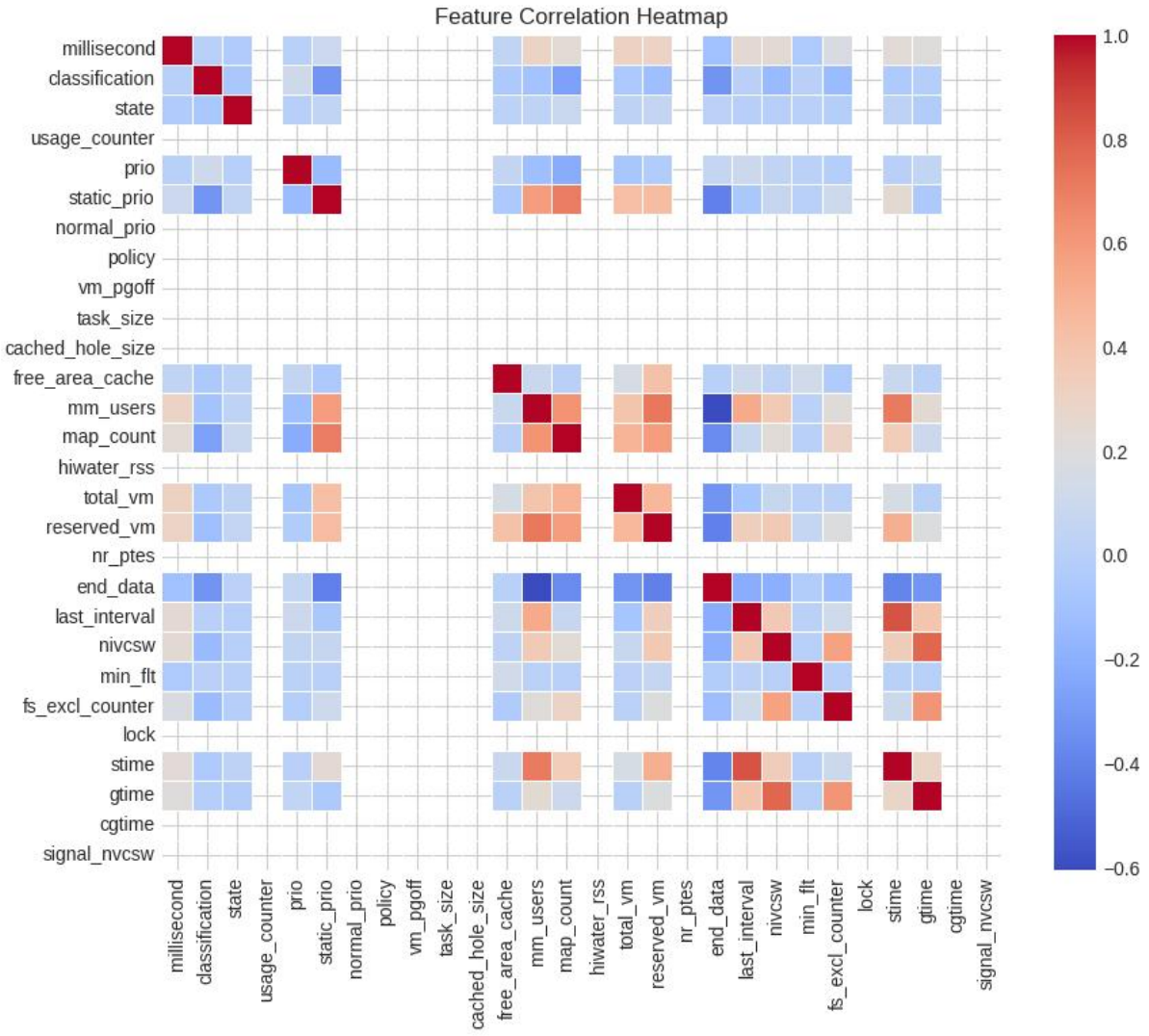

Figure 4. Feature Correlation Heatmap plot

Figure 4 illustrates the feature correlation heatmap will give a general perspective of the relationship between different metrics on the system level, giving the presence or absence of relations in the data. The correlation between most of the variables is weak or insignificant, meaning that a lot of features act independently. But there are sets of tighter relationships between memory related properties like free space cache, mm user, count of maps, total number of virtual machines and reserved number of virtual machines, which indicate that these variables co-vary in response to system activity. Some of the scheduling and process-based characteristics, such as prio, static prio and normal prio, also exhibit moderate correlation patterns. The negative correlation is observed in a few areas, particularly in areas where resource consumption and pressure on the memory are in opposite relationship. In general, the heatmap shows foci of significant correlation and affirms that the data is mostly made up of low-dependency features, which can be used to determine which variables

are most significant to modelling or feature-selection tasks.

Table 1. Naive Bayes Performance

Naïve Bayes	Precision	Recall	F1-score	Support
0	0.77	0.35	0.48	10000
1	0.58	0.89	0.70	10000
Accuracy			0.62	20000
Macro avg	0.67	0.62	0.59	20000
Weighted avg	0.67	0.62	0.59	20000

The results of table 1 Naive Bayes depict the conclusions of an uneven performance of the two classes. In case of class 0, the model has good precision (0.77) and very low recall (0.35) meaning that it can classify correctly very few true instances of this class. Comparatively, class 1 is less precise (0.58) but with a significantly stronger recall (0.89), that is, this model is more likely to gather the majority of positive cases but it yields more false positives. This imbalance is manifested in the overall accuracy of 0.62. The macro and weighted averages have a similar level of precision (0.67), recall (0.62), and F1-score (0.59), indicating that the model is weak at being consistent in performance across the classes. All in all, the report shows that Naive Bayes biases in favor of class 1 but does poorly on class 0 hence performance is mediocre with regards to accuracy although they show significant variance between the classes.

Table 2. SVM Bayes Performance

SVM	Precision	Recall	F1-score	Support
0	0.98	0.99	0.99	10000
1	0.99	0.98	0.99	10000
Accuracy			0.99	20000
Macro avg	0.9940	0.9939	0.9939	20000
Weighted avg	0.9940	0.9939	0.9939	20000

Table 2 shows the SVM model delivers exceptionally strong and well-balanced performance across both classes, with precision, recall, and F1-scores all hovering around 0.98–0.99. Class 0 achieves a recall of 0.99 and precision of 0.98, while class 1 shows the reverse pattern with a precision of 0.99 and recall of 0.98, indicating the model handles both classes almost equally well and makes very few misclassifications. The overall accuracy of 0.99 reflects this consistency, and both the macro and weighted averages confirm the model's stability with identical values across metrics. These results show that SVM captures the underlying patterns of the dataset extremely effectively, achieving near-

perfect performance with minimal performance differences between the two classes.

Table 3. KNN Performance

KNN	Precision	Recall	F1-score	Support
0	0.9993	0.9991	0.9992	10000
1	0.9991	0.9993	0.9992	10000
Accuracy			0.9992	20000
Macro avg	0.9992	0.9992	0.9992	20000
Weighted avg	0.9992	0.9992	0.9992	20000

Table 3 shows KNN model demonstrates very high and virtually perfect performance on all measures of evaluation, precision, recall, and F1-scores were above 0.999 on both classes. Class 0 and class 1 show hardly any difference in their results meaning that the model differentiates the two categories with striking precision and with very minimal mistakes. This good performance is verified by the actual accuracy of 0.9992 and the macro and weighted averages are identical which indicates that the model is stable and balanced despite the distribution of classes. In general, the analysis indicates that KNN is a very suitable algorithm to work with this dataset since the classification rates were close to perfection, and the algorithm showed all the desired indicators in a high degree of reliability.

Table 4 Logistic Regression Model Performance

Logreg	Precision	Recall	F1-score	Support
0	0.91	0.86	0.88	10000
1	0.87	0.91	0.89	10000
Accuracy			0.89	20000
Macro avg	0.8923	0.8911	0.8910	20000
Weighted avg	0.8923	0.8911	0.8910	20000

Table 4 shows the Logistic Regression model performance which is a good and fairly balanced results on the two classes, where the precision, recall, and F1-scores are largely in the upper-80%. In class 0, the model attains a precision of 0.91 but lower recall of 0.86, which indicates that most negative cases are detected by the model but others are not. Class 1 demonstrates the converse with a very marginally smaller precision of 0.87 but the stronger recall of 0.91 making it clear that the model is capable of capturing major positive cases with few false positives. The total accuracy of 0.89 is an indication of this combination of strengths and trade-offs. The macro and weighted averages are about 0.89 on all measures and indicates that the model is performing well with no significant bias in any of the classes. On the whole, the Logistic Regression gives reliable findings, but not as good as the best models in the comparison.

Table 5. Overall Model Comparison Model Performance with Parameters like Accuracy

	Accuracy	Precision	Recall	F1	ROC_AUC
NaiveBayes	0.62	0.58	0.89	0.70	0.69
SVM	0.9939	0.9990	0.9988	0.9939	0.9985
KNN	0.9992	0.9991	0.9993	0.9992	0.9998
LogReg	0.89	0.87	0.91	0.89	0.95

The comparison table 5 of the performance reveals that there are clear variations in the manner in which the individual machine learning models approach the task of classification. Naive Bayes is the worst in overall accuracy and in terms of precision, its accuracy is 0.62 with low precision, but it has a fairly high recall that means it finds a good number of positive cases at the expense of a very large number of false alarms. Conversely, SVM and KNN are particularly impressive because they are the highest performing in all the metrics, with a high accuracy of above 0.99 and also high precision, recall, and F1 and ROC-AUC values. These findings indicate that the two models are very useful in the classification of classes with minimal errors. Logistic Regression does not perform as well as SVM and KNN, having a high accuracy (0.89) and balanced precision and recall, which is why it can be considered a good but inferior choice. All in all, this table indicates that more sophisticated non-linear methods such as SVM and KNN represent the most powerful methods to detect the data, whereas simpler methods, such as Naïve Bayes do not represent the underlying complexity.

4. Conclusion

Comparative analysis of machine learning algorithms indicates the increased significance of data-driven solutions to mitigate the contemporary cybersecurity risks, particularly the ones caused by sophisticated malwares and spam-based vectors of infiltration. The disparities in operation of the reviewed models show the flaws of the conventional probabilistic classifiers such as Naive Bayes that were less effective in their capacity to handle the complicated patterns of behavior that prevails in the modern malware. In comparison, more complex algorithms like Support Vector Machines, and K-Nearest Neighbors achieved close-optimal scores in accuracy, recall and F1-scores, which suggests their capacity to approximate nonlinear decision boundaries, and which consider minor anomalies in large data platforms. The consistency of the results obtained in the Logistic Regression was average, which proves the necessity of applying the algorithms that should align with the dynamics of cyber-attacks. In general, the findings confirm the argument that machine learning can provide the malware detection system with a solid methodological foundation due to its ability to provide adaptive learning, improved generalization, or resistance to novel or obfuscated threats. Advanced ML models are important to the formation of proactive, scalable, and analytically reasonable cybersecurity systems due to the increasing application of evasive tactics

by cyber adversaries and the vast scale of attack surfaces possible with the aid of spam email messages.

5. Referencs

1. Nag, A., Hassan, M. M., Das, A., Sinha, A., Chand, N., Kar, A., ... & Alkhayyat, A. (2024). Exploring the applications and security threats of Internet of Thing in the cloud computing paradigm: A comprehensive study on the cloud of things. *Transactions on Emerging Telecommunications Technologies*, *35*(4), e4897.
2. Villegas, A., & Chen, L. (2011). PPSAM: Proactive PowerShell Anti-Malware: Customizable Comprehensive Tool to Supplement Commercial AVs. In Proceedings of the International Conference on Security and Management (SAM) (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
3. Acharya, J., Chaudhary, A., Chhabria, A., & Jangale, S. (2021, May). Detecting malware, malicious URLs and virus using machine learning and signature matching. In *2021 2nd International Conference for Emerging Technology (INCET)* (pp. 1-5). IEEE.
4. Gopinath, M., & Sethuraman, S. C. (2023). A comprehensive survey on deep learning-based malware detection techniques. *Computer Science Review*, *47*, 100529.
5. Kumar, S., Ahlawat, P., & Sahni, J. (2024). IOT malware detection using static and dynamic analysis techniques: A systematic literature review. *Security and Privacy*, *7*(6), e444.
6. Butt, K. K., Yousif, M., Sumra, I. A., Qazi, A., & Khan, S. (2025). Blockchain in the Digital Age: Challenges, Opportunities, and Future Trends. *Journal of Computing & Biomedical Informatics*, *8*(02).
7. Abubakar, M., Sattar, A., Manzoor, H., Farooq, K., & Yousif, M. (2025). Iiot: An infusion of embedded systems, tinyml, and federated learning in industrial iot. Journal of Computing & Biomedical Informatics, 8(02).
8. Vlahou, A., Hallinan, D., Apweiler, R., Argiles, A., Beige, J., Benigni, A., ... & Vanholder, R. (2021). Data sharing under the General Data Protection Regulation: time to harmonize law and research ethics?. *Hypertension*, *77*(4), 1029-1035.
9. 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.
10. Hakami, A. (2024). Strategies for overcoming data scarcity, imbalance, and feature selection challenges in machine learning models for predictive maintenance. *Scientific Reports*, *14*(1), 9645.
11. Ferdous, J., Islam, R., Mahboubi, A., & Islam, M. Z. (2025). A Survey on ML Techniques for Multi-Platform Malware Detection: Securing PC, Mobile Devices, IoT, and Cloud Environments. *Sensors (Basel, Switzerland)*, *25*(4), 1153.
12. Sabbah, A., Jarrar, R., Zein, S., & Mohaisen, D. (2025). Empirical Evaluation of Concept Drift in ML-Based Android Malware Detection. arXiv preprint arXiv:2507.22772.
13. Kozák, M., Jurečˇek, M., Stamp, M., & Troia, F. D. (2024). Creating valid adversarial examples of malware. *Journal of*

Computer Virology and Hacking Techniques, *20*(4), 607-621.

14. Louthánová, P., Kozák, M., Jurečˇek, M., Stamp, M., & Di Troia, F. (2024). A comparison of adversarial malware generators. *Journal of Computer Virology and Hacking Techniques*, *20*(4), 623-639.
15. Pawlicki, M., Pawlicka, A., Kozik, R., & Choraś, M. (2025). A meta-survey of adversarial attacks against artificial intelligence algorithms, including diffusion models. *Neurocomputing*, 131231.
16. Kayode, B., Adebola, N. T., & Akerele, S. The State of AI-Driven Cybersecurity: Trends, Challenges, and Opportunities. *J Artif Intell Mach Learn & Data Sci 2025*, *3*(2), 2731-2739.
17. Molina-Coronado, B., Mori, U., Mendiburu, A., & Miguel-Alonso, J. (2023). Towards a fair comparison and realistic evaluation framework of android malware detectors based on static analysis and machine learning. *Computers & Security*, *124*, 102996.
18. Ferdous, J., Islam, R., Mahboubi, A., & Islam, M. Z. (2025). A Survey on ML Techniques for Multi-Platform Malware Detection: Securing PC, Mobile Devices, IoT, and Cloud Environments. *Sensors (Basel, Switzerland)*, *25*(4), 1153.
19. Wu, T., Fan, H., Zhu, H., You, C., Zhou, H., & Huang, X. (2022). Intrusion detection system combined enhanced random forest with SMOTE algorithm. *EURASIP Journal on Advances in Signal Processing*, *2022*(1), 39.
20. Fan, R., Tian, A., Li, Y., Gu, Y., & Wei, Z. (2025). Research Progress on Machine Learning Prediction of Compressive Strength of Nano-Modified Concrete. *Applied Sciences*, *15*(9), 4733.
21. Guo, W., Du, W., Yang, X., Xue, J., Wang, Y., Han, W., & Hu, J. (2025). MalHAPGNN: An enhanced call graph-based malware detection framework using hierarchical attention pooling graph neural network. *Sensors*, *25*(2), 374.