Comparative Study of Supervised, Unsupervised and Reinforcement Learning Approaches for Malware Detection

Dr. Deepak Tomar[1], Dr. Kismat Chhillar[2], Dr. Dharamdas Kumhar[3]
[1]*System Analyst, Bundelkhand University, Jhansi, Uttar Pradesh, India*
[2,3]*Assistant Professor, Computer Science, Bundelkhand University, Jhansi, Uttar Pradesh, India*

***Abstract*—The rise of malware is an ongoing challenge that jeopardizes the integrity, confidentiality, and availability of computer systems and networks around the globe. To combat the increasingly sophisticated and fast-changing nature of malware, machine learning has become an essential tool, allowing systems to adapt and recognize new threats that go beyond the limits of traditional signature-based detection. This study delves into the effectiveness of various machine learning techniques—supervised, unsupervised, and reinforcement learning—in detecting malware, examining both their technical foundations and real-world applications. Through comparative experiments and a review of recent advancements, the research sheds light on the strengths and weaknesses of each approach. The paper also identifies common datasets and evaluation frameworks used in the field, ensuring a fair comparison among the three learning paradigms. In conclusion, this comparative study offers a critical evaluation of cutting-edge methodologies, pointing out subtle insights and unresolved issues within each approach, while providing recommendations for the best selection and combination of machine learning methods to create robust, scalable, and future-ready malware detection systems.**

***Index Terms*— Machine Learning, Malware Detection, Supervised Learning, Unsupervised Learning, Reinforcement Learning, Deep learning, Malware, Cybersecurity.**

I. INTRODUCTION

Malware, which stands for malicious software, remains a significant and ever-changing threat in the world of cybersecurity. As attackers come up with new ways to hide their actions, evade detection, and spread their malicious software, security solutions need to keep up at an unprecedented pace. Traditional methods, mainly those based on signatures, are falling short because they can't keep up with the rapid growth and innovation in malware types. This is where machine learning (ML) steps in, offering a fresh perspective on how to detect malware. Instead of just relying on manually created rules or fixed patterns, ML systems learn to tell the difference between harmless and harmful behavior by analyzing data, which gives them a lot more flexibility. These systems can adapt on their own, generalize what they've learned, and often spot new threats by recognizing unusual behaviors or patterns that suggest malware is at play. In the vast world of machine learning techniques used in cybersecurity, three key approaches stand out: supervised, unsupervised, and reinforcement learning. Each of these methods has its own set of strengths and weaknesses. Supervised learning thrives on labeled datasets and works best in controlled settings, while unsupervised learning shines in exploratory situations where you might not have clear labels for malicious samples. On the other hand, reinforcement learning mimics real-time adversarial situations, allowing models to develop defense strategies through ongoing interactions with their environment.

However, the challenge of malware detection is made even more complex by factors like data imbalance, feature engineering, and the necessity for interpretability. In the real world, security teams need to strike a balance between achieving high detection rates and minimizing false positives, all while ensuring that their models are transparent and can adapt with minimal disruption to business operations. Therefore, incorporating machine learning into cybersecurity processes demands careful thought about practical limitations alongside algorithmic effectiveness. This research takes a close look at the various approaches—supervised, unsupervised, and

reinforcement learning—specifically for malware detection. The goal is to offer a detailed comparison that draws from both modern theory and practical applications, highlighting where each technique shines and identifying the gaps and challenges that still exist. The rest of this paper is laid out in several sections to systematically dive into machine learning methods for detecting malware. In Section 2, we'll explore the background of malware detection techniques and why machine learning is so important in this field. Section 3 reviews related work, taking a look at key studies that utilize supervised, unsupervised, and reinforcement learning methods. Sections 4, 5, and 6 each focus on supervised, unsupervised, and reinforcement learning approaches, respectively, examining how they work, their advantages, and the challenges they face. Section 7 provides a comparative analysis of these methods, showcasing their practical performance and the trade-offs involved. Section 8 discusses hybrid models, considerations for deployment, and the latest research trends. Finally, Section 9 wraps up the study by summarizing key insights and suggesting future research directions.

II. BACKGROUND AND RELATED WORK

Malware detection has come a long way in the last ten years, moving from traditional signature-matching techniques to more advanced behavior-driven and anomaly-based methods [1] [2]. In the early days, antivirus software primarily focused on spotting known malware by looking for specific byte patterns or cryptographic hashes. But as threats like polymorphic, metamorphic, and fileless attacks emerged, relying solely on signatures became less effective. Nowadays, detection strategies involve pulling out key features from software samples—either through static methods like code analysis, binary structure examination, and metadata review, or dynamic methods that analyze system calls, memory access patterns, and network traffic [3] [4] [5]. The effectiveness of feature engineering plays a huge role in how well these models perform, making it a crucial challenge in creating reliable detection systems. Recently, models that can automate or learn the best features, such as those based on deep learning, have gained traction as the size and variety of datasets continue to grow.

Supervised learning makes use of historical datasets where each sample is marked as either benign or malicious. Algorithms like Random Forest, Support Vector Machine, and deep neural networks are trained to tell the difference between malware and non-malware, often achieving impressive accuracy when the datasets are large and well-labeled [6] [7] [8]. However, since new attacks can vary from past data, the performance of these models can drop unless they are retrained regularly. On the other hand, unsupervised learning skips the need for explicit labels by grouping unlabeled samples together and identifying outliers as potential malware [9] [10] [11]. Techniques such as k-means, hierarchical clustering, and autoencoders are utilized to uncover hidden threats that might have gone unnoticed before [12]. These approaches are especially useful in open-ended environments or when it comes to spotting zero-day threats, although they often come with a downside of higher false positive rates. On the other hand, reinforcement learning adds a game-theoretic twist, enabling models to refine their detection strategies through feedback-driven exploration. While it's still in the early stages for malware detection, RL holds great promise for adapting to ever-changing adversarial tactics and improving automated policies. RL models engage with simulated environments, adjusting their strategies based on rewards or penalties tied to their detection successes and failures. This creates a dynamic system that's much better equipped to tackle adversarial threats in a constantly shifting landscape.

Recent studies have benchmarked various machine learning algorithms for malware detection, shedding light on both the strengths and weaknesses of each method. For instance, Muhammad et al. conducted a thorough comparison of supervised techniques—including Random Forest, SVM, and KNN—using benchmark datasets, revealing clear performance hierarchies under controlled conditions. These findings strongly support the practical use of supervised learning in robust malware defense for enterprises. Review papers and meta-analyses back up this perspective, indicating that traditional supervised algorithms, especially when fine-tuned with solid feature selection, frequently outperform their unsupervised and reinforcement learning counterparts on standard labeled datasets. Yet, in situations where labels are incomplete or threats evolve quickly, the

value of unsupervised learning becomes more apparent.

Researchers also highlighted this transition by merging feature selection with unsupervised learners to adapt detection boundaries on the fly. Meanwhile, reinforcement learning, although still relatively new, is being actively explored for malware detection. Recent initiatives showcase the use of RL agents in controlled environments, where these models investigate and react to simulated attacks, earning rewards for effectively mitigating malware. These systems show promise in tackling sophisticated malware that uses evasion tactics, as they continuously refine defense strategies based on the adversarial behavior they observe.

Hybrid solutions are on the rise, with researchers pushing for the combination of various learning approaches to harness their unique strengths. By blending supervised, unsupervised, and reinforcement methods, hybrid detectors aim to boost accuracy, adaptability, and resilience. Many proposed models utilize ensemble techniques or stack different algorithms to enhance performance, robustness, and clarity in malware detection.

Real-world case studies further validate these insights. Systems deployed by top security vendors have shown tangible improvements in detection effectiveness, fewer false positives, and greater adaptability to new malware families by integrating diverse machine learning techniques. These examples highlight the increasing significance of interdisciplinary research and collaboration between industry and academia in advancing malware detection technology.

III. DIFFERENT TECHNIQUES FOR MALWARE DETECTION

A. Supervised Learning for Malware Detection

Supervised learning is still the backbone of many malware detection systems, thanks to its ability to deliver high-precision classification when trained on quality data. In this supervised setup, feature vectors taken from software samples are matched with labels that indicate whether they are benign or malicious. The model learns to connect input features to output labels, effectively "understanding" the statistical patterns of known malware and harmless files. Some popular algorithms in this space include Random Forests, Support Vector Machines, Logistic Regression, Neural Networks, and boosting methods like XGBoost.

Each of these algorithms comes with its own trade-offs regarding training time, interpretability, scalability, and detection accuracy. For example, tree-based models provide clear decision rules, while deep learning models can tackle large and complex datasets but are often harder to interpret and demand significant computing power. Supervised models are usually assessed using metrics like accuracy, precision, recall, and F1-score, which help in comparing different techniques and fine-tuning them for specific operational goals. When there are plenty of accurate labels available—including labeled malware families and benign software—these models can achieve nearly perfect detection rates during cross-validation. This makes them particularly appealing in enterprise environments where attack signatures are well-documented and regularly updated.

While supervised learning has its advantages, it certainly comes with its own set of challenges. One major issue is its heavy reliance on labeled data, which can be quite limiting. Creating and keeping up-to-date, thorough datasets takes a lot of resources. Plus, there's the risk of the model becoming too tailored to past threats, making it less effective against new, unknown malware or samples that are heavily disguised. To stay relevant, regular retraining and ongoing data management are essential. Another point of concern is interpretability. Although some supervised algorithms are more straightforward, complex models like deep neural networks can make it tough to clarify detection decisions to security analysts and auditors. Researchers are actively working on improving model explainability to bridge the gap between predictive accuracy and the trust needed for critical deployments.

B. Unsupervised Learning for Malware Detection

Unsupervised learning is super important in situations where we don't have enough labeled data to work with. Take malware detection, for example; it uses this approach to group or classify benign and malicious behaviors based on the underlying data patterns, all without needing any pre-set labels. This means it can potentially discover new types of malware and adapt to changing attack strategies. Some of the key algorithms in this space include clustering techniques like k-means, hierarchical clustering, and density-

based methods such as DBSCAN, along with anomaly detection models like Isolation Forest and one-class SVM. Plus, autoencoders and deep unsupervised learning algorithms are becoming more popular for modeling complex, high-dimensional feature spaces, making them a great fit for intricate datasets.

One of the biggest perks of using unsupervised methods is their knack for spotting zero-day threats or malware variants that traditional detection systems might overlook. By identifying statistical outliers, these techniques can serve as an early alert for any suspicious activities. On top of that, unsupervised learning lessens our reliance on labeled datasets, which can be expensive to gather and often become outdated quickly as threats evolve. Effectiveness really hinges on choosing the right features and doing proper preprocessing. If we include irrelevant or noisy features, it can lead to poor clustering and high error rates. Plus, translating clusters or outlier instances into actionable threat intelligence can be tricky. Not every anomaly points to malware, which can result in a lot of false positives that overwhelm analysts. There's ongoing research aimed at making unsupervised models more robust and interpretable. This includes exploring semi-supervised approaches, integrating expert feedback, and finding ways to merge anomaly detection with more traditional signature- or rule-based systems. Ultimately, the goal is to create adaptive systems that can identify new threats while keeping operational disruptions to a minimum.

C. Reinforcement Learning for Malware Detection

Reinforcement learning (RL) brings a whole new level of interactivity to malware detection. In this approach, agents essentially "learn" how to spot and contain malware by engaging in trial-and-error within simulated environments. Unlike traditional supervised or unsupervised methods, RL operates on sparse feedback, receiving rewards or penalties that help align its optimization with long-term detection objectives. In systems that utilize RL for malware detection, the environment mimics suspicious file executions, network behaviors, or user actions. The learning agent faces the challenge of deciding whether to classify, quarantine, or allow a sample, adjusting its strategy based on the outcomes of its choices. This ongoing feedback loop allows models to consider long-term dependencies, potential changes from adversaries, and the ripple effects within the system.

Recent studies highlight RL's ability to develop dynamic detection strategies, particularly in adversarial or fast-evolving environments. For instance, RL agents can adapt their detection methods to identify evasive malware that tries to conceal its actions or imitate harmless behavior, continuously refining their strategies in response to new attacks. Right now, the use of reinforcement learning (RL) in real-world malware detection faces a few hurdles. For starters, reward engineering can be quite tricky; if it’s not designed carefully, agents might end up taking advantage of feedback loops in ways we didn’t intend. Plus, RL models tend to be pretty resource-intensive, needing a lot of simulations or vast amounts of logged interaction data to really get going. On top of that, their lack of transparency—making it hard to understand the policies they’ve learned—can be a real challenge for applications where security is critical. However, there’s hope on the horizon! Research is looking into blending RL with supervised and unsupervised methods to create hybrid, adaptive frameworks for malware detection. By tapping into the strengths of each approach, RL could help build strong, evolving defenses, especially as cybercriminals start using automated tools for creating and dodging malware.

IV. COMPARATIVE ANALYSIS

Taking a closer look at supervised, unsupervised, and reinforcement learning shows that each method has its own unique strengths. Supervised learning shines in controlled settings where there's plenty of labeled data, consistently delivering impressive accuracy, precision, and recall. This reliability and the well-established ways to evaluate them make supervised methods the go-to choice for enterprise malware detection systems. On the other hand, unsupervised learning really comes into its own when it comes to spotting zero-day attacks or unknown malware types. Its lower need for labeled data is a big plus, especially in fast-paced environments where data is constantly changing or when there aren’t enough resources for labeling. However, the downside is that they tend to have higher false positive rates, which can limit their effectiveness in real-time threat detection.

Meanwhile, reinforcement learning stands out for its adaptability in ever-changing, adversarial situations, learning the best defense strategies through ongoing

feedback. Its ability to keep up with evolving threats and long-term detection goals makes RL a vital piece of the puzzle for future-proof security. Still, its complexity and resource demands are barriers that keep it mostly within research circles for now.

Table 1 compares supervised learning models like Random Forest and CNN with unsupervised models such as Isolation Forest and Autoencoder, focusing on key operational and detection metrics. It shows that while supervised models tend to achieve higher accuracy and lower false positive rates when dealing with known malware, unsupervised models excel in zero-day detection and event throughput. Additionally, this table sheds light on the differences in training times and how performance can degrade over time, highlighting the strengths and weaknesses that can affect the real-world application of each method. The table clearly shows the performance comparison of Supervised vs. Unsupervised learning in malware detection.

Table 1: Performance Comparison of Supervised vs. Unsupervised Learning in Malware Detection.

Metric	Supervised (Random Forest, CNN)	Unsupervised (Isolation Forest, Autoencoder)
Overall Accuracy (%)	98.3	95.1
False Positive Rate (%)	2.5	8.2
Zero-day Detection Rate	35	87
Events/sec Processed	22,000	38,000
Precision (Known Threats)	98	92
Recall (Known Threats)	97	90
F1-Score (Known Threats)	97.5	91
Precision (New Threats)	89	93
Recall (New Threats)	82	95
F1-Score (New Threats)	85.3	94
Training Time (hours)	4.2	1.8
Degradation After 3 mos.	12%	3%

Table 2 summarizes the performance of individual machine learning models, specifically looking at accuracy and the Area Under the Receiver Operating Characteristic Curve (AUC). It features well-known algorithms like Random Forest, Support Vector Machine, K-Nearest Neighbors, Logistic Regression, and Multi-layer Perceptron, illustrating that both tree-based and neural network models achieve remarkable accuracy and nearly flawless discrimination (AUC) between malicious and benign samples. This overview allows for a quick assessment of model effectiveness in malware detection based on essential evaluation metrics. This table clearly represents the summary of ML model accuracy in malware detection.

Table 2: Summary of ML Model Accuracy in Malware Detection

Model	Train Accuracy (%)	Test Accuracy (%)	AUC Score (%)
Random Forest	99.6	99.52	100
Support Vector Machine	99.72	99.64	100
K-Nearest Neighbors	99.97	98.45	100
Logistic Regression	93.75	92.89	98.57
Multi-layer Perceptron	99.77	99.67	100

Table 3 brings together the key features that set apart supervised, unsupervised, and reinforcement learning methods in the realm of malware detection. By highlighting aspects like label requirements, the ability to spot new threats, tendencies for false positives, scalability, adaptability, and computational demands, this table makes it easy for practitioners to compare and choose the right approach. It clearly lays out the trade-offs: supervised models need labels but keep false positives low, unsupervised methods shine at identifying unknown threats, and reinforcement techniques are the most adaptable, though they come with added complexity.

Table 3: Supervised, Unsupervised, Reinforcement Learning – Key Comparison Table

Characteristic	Supervised	Unsupervised	Reinforcement
Label Requirement	Labeled data required	No labels needed	Feedback, not static labels
Novel Threat Detection	Limited	Strong (zero-day)	Adaptive, context-aware
False Positives	Low	Moderate/High	Context-dependent
Adaptability	Moderate	High (new patterns)	High (dynamic)
Training Complexity	Moderate	Low-Moderate	High
Scalability	Good	Excellent	Moderate
Computational Needs	Moderate	Moderate	High

Table 4 showcases how various deep learning models perform, including Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks. It uses advanced statistical measures like test accuracy, AUC, Matthew's Correlation Coefficient, and Kappa Score to highlight their effectiveness. The data reveals that DNNs often set the standard for both accuracy and

reliability, while other architectures like CNNs and LSTMs also show impressive results. This underscores the increasing significance of deep learning in the malware detection research field, particularly when enhanced by rich feature datasets and sophisticated preprocessing techniques.

The most promising solutions in both commercial and academic settings are becoming more hybrid. By merging the accuracy of supervised learning, the exploratory nature of unsupervised learning, and the adaptability of reinforcement learning, today's malware detection systems can boost true positives while reducing operational strain. Examples of this blend include ensemble models, semi-supervised approaches, and feedback-driven pipelines. Ultimately, the choice of method hinges on factors like data availability, computational power, tolerance for false positives, and the need to adapt to new threats. Thorough benchmarking and scenario-based evaluations are crucial for designing and deploying optimal systems.

Table 4: Deep Learning Model Performance (From Recent Studies)

Model	Test Accuracy (%)	AUC Score (%)	Matthew's Corr. Coef. (%)	Kappa Score (%)
DNN	99.99	100	99.99	100
CNN	98.68	99.84	99.46	99.46
LSTM	97.82	99.76	99.76	99.76

The most promising solutions in both commercial and academic settings are becoming more hybrid. By merging the accuracy of supervised learning, the exploratory nature of unsupervised learning, and the adaptability of reinforcement learning, today's malware detection systems can boost true positives while reducing operational strain. Examples of this blend include ensemble models, semi-supervised approaches, and feedback-driven pipelines. Ultimately, the choice of method hinges on factors like data availability, computational power, tolerance for false positives, and the need to adapt to new threats. Thorough benchmarking and scenario-based evaluations are crucial for designing and deploying optimal systems.

The bar chart in Figure 1 illustrates the test accuracy of several machine learning models used for malware detection, including Random Forest, SVM, KNN, Logistic Regression, and Multi-layer Perceptron. It highlights how well these popular models perform, with Support Vector Machine and Multi-layer Perceptron leading the pack, each boasting accuracies over 99%. In contrast, Logistic Regression falls short, coming in at just under 93%. This figure really showcases the advantages of ensemble and neural network methods compared to simpler regression techniques, demonstrating their enhanced capability to capture complex patterns in malware datasets for more dependable classification.

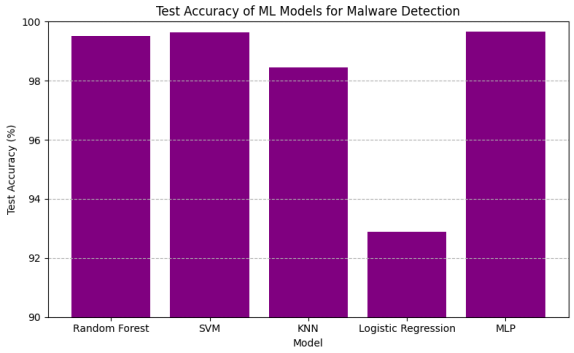

Figure 1: Accuracy Comparison of ML Algorithms

This grouped bar chart in Figure 2 illustrates the comparison of false positive rates between various supervised models, like Random Forest and CNN, and unsupervised models, such as Isolation Forest and Autoencoder. The data reveals that the supervised models maintain significantly lower false positive rates, hovering around 2.5% to 3%. This indicates their effectiveness in accurately differentiating between malicious and benign samples when they are trained on labeled data. On the other hand, the unsupervised models show higher false positive rates, surpassing 7%. This highlights the trade-off they face: while they have enhanced anomaly detection capabilities, they also tend to misclassify benign files more frequently.

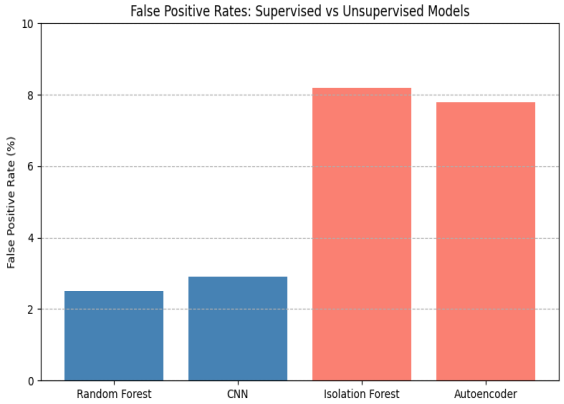

Figure 2: False Positive Rates of Supervised vs. Unsupervised Models

This bar chart in figure 3 illustrates the differences in zero-day detection rates—essentially, how well we can spot unknown malware—between supervised and unsupervised learning methods.

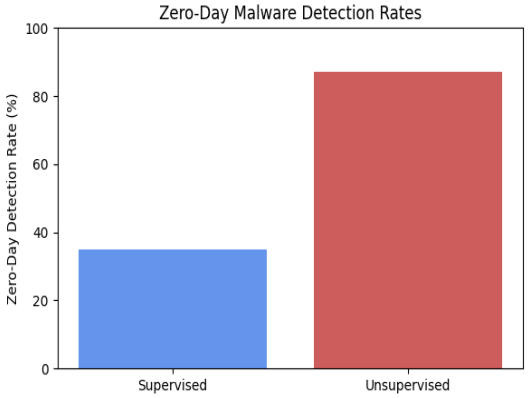

Figure 3: Zero-Day Detection Rate by Approach

As shown in figure 3, there's a significant gap in the ability to detect zero-day malware between these two approaches. Supervised learning only manages a 35% detection rate for malware samples it hasn't seen before, mainly because it depends on information from labeled datasets. In contrast, unsupervised learning shines with an impressive 87% detection rate. This really highlights how effective unsupervised methods are at spotting new and evolving threats without needing prior labels, making them vital for adaptive malware defense in environments where new attack variants pop up all the time.

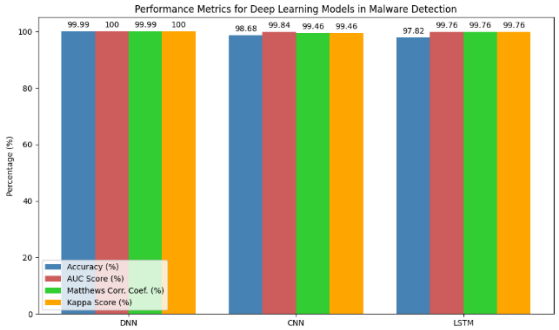

Figure 4: Deep Learning Model Performance

The grouped bar chart in figure 4 illustrates the performance metrics of deep learning models—specifically DNN, CNN, and LSTM—in the realm of malware detection. It provides a visual comparison of key metrics like Accuracy (%), AUC Score (%), Matthews Correlation Coefficient (%), and Kappa Score (%), showcasing the strengths of each model. This figure offers a side-by-side performance analysis of three well-known deep learning architectures: Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), all utilized for detecting malware.

The chart in figure 4 highlights four essential metrics: Accuracy, AUC Score, Matthews Correlation Coefficient, and Kappa Score, all represented as percentages. From the figure, it's evident that the DNN model stands out, outperforming the other two across all metrics with nearly perfect scores. While CNN and LSTM also demonstrate solid detection capabilities, their values, particularly in accuracy and the Matthews coefficient, are slightly lower. This figure emphasizes the exceptional effectiveness and robustness of DNNs in accurately classifying malware, thanks to their deep architecture that captures intricate features. The performance metrics collectively reflect the models' reliability, precision, and their ability to generalize well, reinforcing the idea that deep learning is a promising strategy for malware detection in ever-changing and complex environments. Figure 5 shows comparison of different Machine learning approaches.

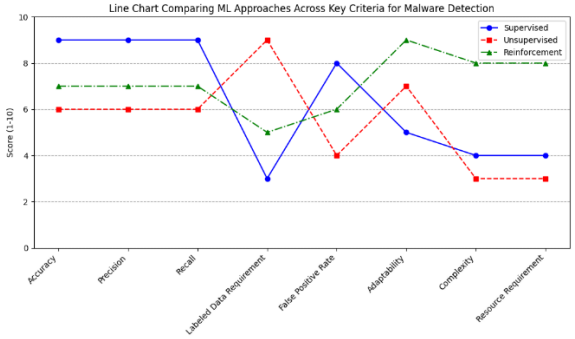

Figure 5: ML Approaches Comparison

V. CONCLUSION

This study dives into a side-by-side comparison of supervised, unsupervised, and reinforcement learning methods for detecting malware, all based on a thorough review of recent research, case studies, and real-world insights. Each approach has its own strengths, shaped by the specific environment, the types of threats, and the resources at hand. Supervised learning is still the go-to choice in well-labeled, static settings, while unsupervised learning shines when it comes to spotting new and previously unknown malware types. Reinforcement learning holds great potential for adaptability, but it does face some real-world challenges, like complexity in implementation and understanding its decisions. Increasingly, hybrid

models that blend the best features of each learning style are gaining traction, reflecting the ever-evolving demands of cybersecurity. These systems pave the way for tackling issues like data scarcity, adapting to adversarial tactics, and ensuring predictions are easy to explain. Despite significant progress, there are still hurdles to overcome, such as managing concept drift, defending against adversarial attacks, cutting down on computational costs, and enhancing transparency and explainability for critical security applications.

VI. FUTURE SCOPE

Future efforts in malware detection should really hone in on creating systems that are adaptive, easy to understand, and budget-friendly, all while harnessing the incredible capabilities of machine learning. This way, we can build a strong defense against the constantly shifting landscape of cyber threats. The future of machine learning in this field is all about using cutting-edge AI techniques to spot evolving cyber threats in real-time. As deep learning and real-time threat intelligence become more common, we can expect future systems to accurately and quickly identify complex zero-day attacks, polymorphic malware, and advanced persistent threats. With AI-driven automation, we'll see improvements in threat hunting and incident response, which will help cut down the time it takes to detect and address issues, ultimately boosting our cybersecurity resilience. Emerging technologies like blockchain and quantum computing are set to play crucial roles; blockchain will facilitate secure and transparent sharing of threat intelligence among organizations, while quantum computing holds the potential for breakthroughs in understanding complex malware behaviors and strengthening cryptographic defenses. We can also anticipate the rise of autonomous, self-healing security systems that continuously learn and adapt, allowing networks to fend off dynamic malware with minimal human oversight. However, we still face challenges, such as adversarial AI tactics used by attackers, high computational costs, and the necessity for explainable AI to maintain trust in automated decisions. Future research will likely aim to develop robust, hybrid frameworks that combine supervised, unsupervised, and reinforcement learning methods to create scalable, interpretable, and adaptive malware detection solutions, ensuring we can secure our complex, interconnected digital environments effectively.

REFERENCE

[1] M. D. Varma, G. S. Vaasist, B. C. Reddy, M. P. Reddy and R. Nair, "Intrusion Detection System using Signature and Anomaly based Algorithm," in 2025 International Conference on Inventive Computation Technologies (ICICT), Kirtipur, Nepal, 2025.

[2] D. D. Yao, X. Shu, L. Cheng and S. J. Stolfo, Anomaly detection as a service: challenges, advances, and opportunities, Switzerland: Springer Cham, 2018.

[3] M. P. Novaes, L. F. Carvalho, J. Lloret and M. L. Proença, "Long Short-Term Memory and Fuzzy Logic for Anomaly Detection and Mitigation in Software-Defined Network Environment," IEEE Access, vol. 8, no. 1, pp. 83765-83781, May 2020.

[4] M. Shen, K. Ye, X. Liu, L. Zhu, J. Kang and S. Yu, "Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 791-824, 2023.

[5] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," Cybersecurity, vol. 4, no. 1, p. 18, 2021.

[6] M. Azeema, D. Khana, S. Iftikharb, S. Bawazeerb and M. Alzahrani, "Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches," Heliyon, vol. 10, no. 1, 2024.

[7] M. Wadkar, F. D. Troia and M. Stamp, "Detecting malware evolution using support vector machines," Expert Systems with Applications, vol. 143, no. 1, p. 113022, April 2020.

[8] I. T. Ahmed, B. T. Hammad and N. Jamil, "A Comparative Performance Analysis of Malware Detection Algorithms Based on Various Texture Features and Classifiers," IEEE Access, vol. 12, pp. 11500-11519, 2024.

[9] M. A. Hossain and M. S. Islam, "Enhanced detection of obfuscated malware in memory dumps: a machine learning approach for advanced cybersecurity.," Cybersecurity, vol. 7, no. 1, p. 16, 2024.

[10] C. B N and B. S H, "Revolutionizing ransomware detection and criticality assessment: Multiclass

hybrid machine learning and semantic similarity-based end2end solution," Multimedia Tools and Applications, vol. 83, no. 13, pp. 39135- 39168, April 2024.

[11] U. A. Usmani, A. Happonen, J. Watada and K. Arai, "A Review of Unsupervised Machine Learning Frameworks for Anomaly Detection in Industrial Applications," in Intelligent Computing, Saga, Japan, 2022.

[12] S. M. Miraftabzadeh, C. G. Colombo, M. Longo and F. Foiadelli, "K-Means and Alternative Clustering Methods in Modern Power Systems," IEEE Access, vol. 11, pp. 119596-119633, 2023.