Milestone Transactions on
Artificial Intelligence
Vol. 01 | Issue. 01 | 2026

RESEARCH ARTICLE OPEN ACCESS

Understanding the Evolving Landscape of Malware Threats Through Cyber Threat Intelligence

D Nagabhushanam*. P S Usha Rani. P Gowthami. S Mohammed Sameer. K Yuvaraj. M Balaji

Department of CSE (IoT, Cyber Security including Block Chain Technology),
Annamacharya Institute of Technology & Sciences (Autonomous),
Tirupati, Andra Pradesh, India.

DOI: **10.5281/zenodo.18525909**
Received: 13 January 2026 / Revised: 22 January 2026 / Accepted: 7 February 2026
Corresponding Author: bhushan.duggi@gmail.com

Abstract – The rapid development of sophisticated forms of malware and ever-changing cyber threats have become a major challenge for cybersecurity globally. Using cyber threat intelligence (CTI), this study aims to provide an overall analysis of the current trend of malware. It has identified various forms of recent attack patterns, behaviors, and evasion techniques of malware. The study has explored various forms of malware activation, propagation, and evasion techniques. It has evaluated their impact on critical infrastructure such as finance, healthcare, and other sectors. Using various case studies, experts, and threat intelligence, this study has demonstrated the importance of timely and accurate threat analysis. A comparative study of various machine learning-based threat classification has been implemented with RF, SVM, and DT algorithms. The performance evaluation of these classifiers shows that the RF classifier performs better than the others. It has achieved an accuracy of 95.57%. Hence, it has been able to show its efficiency while dealing with problems that involve a large number of dimensions, such as cybersecurity. The importance of intelligent detection and mitigation techniques in dealing with malware attacks has been revealed. The importance of international cooperation and collaboration in dealing with malware threats has been demonstrated.

Index Terms – Malware Detection, Cyber Threat Intelligence, Machine Learning, Random Forest, Cybersecurity, Threat Analysis.

I. INTRODUCTION

The rapid evolution of digital technology has brought about a revolution in the modern world, making it possible to have smooth communication, automation, and data transfer between cloud systems, business networks, mobile phones, and IoT devices. However, the digital revolution has also resulted in

an increase in cyber threats, and malware is one of the most critical and alarming threats. Malware, which includes viruses, worms, trojans, ransomware, spyware, and botnets, is used to attack computer systems and steal confidential data, thus compromising critical infrastructures. In recent years, malware, including polymorphic malware, encrypted malware, and fileless malware, has become increasingly sophisticated and has significantly reduced the efficiency of traditional signature and rule-based detection systems [1], [2].

Traditional cybersecurity systems are highly dependent on predefined signatures and heuristic rules for the detection of malicious software. Although these methods are quite effective for known threats, they are not very efficient for the detection of zero-day attacks and unknown variants of malware. With the increasing use of automated tools and artificial intelligence by attackers for developing adaptive malware, the need for intelligent and data-driven approaches for defensive systems has also arisen [1]. In this regard, Artificial Intelligence (AI) and Machine Learning (ML) have shown great promise for improving malware detection, classification, and prediction tasks. ML algorithms are capable of processing large amounts of system logs, network traffic, and executable attributes to discover hidden patterns that can distinguish malicious activities from benign ones [2].

Recent studies have also demonstrated the efficiency of deep learning and hybrid ML paradigms in increasing the accuracy of the detection process. Malware classifiers using AI technologies are capable of automatically learning complex feature representations, thus allowing for early detection of malicious activities. Explainable AI technologies are also being incorporated in the security landscape to increase the transparency of the ML model, particularly for the detection of Advanced Persistent Threats (APTs), which remain undetected for long periods of time. Behavior-based detection techniques are also being implemented for increasing the security of the system, particularly for the detection of ransomware and polymorphic malware, which often modify their code structures while maintaining their behavioral patterns [3].

Despite such advances, there are a number of technical challenges that AI-powered malware detection systems currently encounter. However, the most prominent among these is the adversarial attacks, which are carried out by intentionally manipulating the input data in such a manner that it misleads the ML classifiers. The research carried out in the context of adversarial learning has highlighted the vulnerabilities of the existing malware detection systems, thereby emphasizing the need for developing more robust architectures [4]. Another area of ongoing research is the combination of Cyber Threat Intelligence (CTI) with automated detection systems. CTI offers valuable context about the tactics, techniques, and procedures of attackers, thus improving predictive analysis and proactive defense strategies [5].

Another security complexity that arises in the expansion of IoT ecosystems is that of device heterogeneity, computational resource limitations, and security configurations. Lightweight ML techniques, such as ensemble-based techniques, such as Random Forest, have also been considered for efficient malware detection in resource-constrained environments [6]. Transfer learning techniques are also being considered for zero-day malware detection by leveraging knowledge from previous attacks [7]. Cloud-based infrastructure, where large amounts of security data are generated, also benefits from big data analytics frameworks for malware analysis and monitoring [8].

Feature engineering and optimization are crucial for improving ML-based detection performance. Selecting relevant features reduces computational overhead and enhances classification accuracy. In addition, ensemble learning strategies, which use multiple classifiers, were found to possess better robustness, stability, and fewer false positive rates compared to single classifiers [9]. It has also been found through comparative analysis that ensemble classifiers, such as Random Forests, provide a good balance between accuracy, interpretability, and efficiency, which is desirable for deploying classifiers in real-world environments such as cybersecurity systems [10]. Although considerable progress has been made, many research gaps remain. Malware is constantly evolving and using sophisticated evasion techniques, which result in problems such as concept drift, scalability issues, high false-positive rates, and a lack of model explainability. Real-time detection and resistance to adversarial manipulation are also unsolved problems.

Our contributions are as follows:

- In this paper, we propose an innovative approach for integrating the structured Cyber Threat Intelligence (CTI) data with the machine learning algorithms for the classification of malware, which can enable the context-aware analysis of malware instead of relying on the signature-based malware detection techniques.
- Comparative Evaluation of Lightweight and Robust ML Classifiers: A thorough comparative evaluation of the Random Forest, Support Vector Machine, and Decision Tree classifiers is performed on a large-scale real-world malware dataset. The results show that the Random Forest classifier is the most reliable, with 95.57% accuracy, to deal with high-dimensional heterogeneous cybersecurity data.
- Unlike previous research, the present research aims to provide a structured approach to the data preprocessing step, which includes handling missing values, label encoding, normalization, feature removal, and stratified sampling to ensure the overall performance of the models in the context of the large-scale nature of the data generated by the network devices.
- previous research, the present research aims to provide an overall platform that supports the best performance of the classifier, which will then be used to provide real-time predictions to support the overall proactive approach to cyber-attacks and threats.

II. LITERATURE SURVEY

To fill the research gap of low detection accuracy and high false-positive rates in the traditional malware detection systems, Fahim et al. [11] suggested a comparative malware detection framework. The researchers compared the performance of the Random Forest, Multilayer Perceptron (MLP), and Deep Neural Network (DNN) models on a large dataset available on the Kaggle platform with an optimized pipeline of feature selection and preprocessing. The DNN model succeeded in the highest performance with 99.92% average and a nearly perfect AUC score. The novelty of this work is seen in the fact that the combination of a deep learning architecture with optimized feature engineering demonstrates the high-detection ability. Nevertheless, the paper has drawbacks related to the use of one data set and the fact that DNNs are computationally expensive, which could prevent real-time application and extension to unknown malware.

To address the gap in research with low precision in PE-file malware families classification, Mahato et al. [12] suggested a Feature-Driven Cascade Machine Learning model (MDCML). The framework combines the random forest, bagging, and boosting classifiers in a cascade format and inter-class dispersion (TF-IDF-based) to extract features. The proposed model achieved 98.97% accuracy on the Big-2015 dataset and 95.42% on the Mal-API-2019 dataset. The most important innovation in this work is the cascade strategy that the uncertain samples are increasingly refined on a series of classifiers, enhancing their robustness and multi-class detection. Nonetheless, the multi-stage processing complicated the computations and reliance on opcode-based characteristics, which restricts scalability and applicability in real-time. Ajayi et al. [13] suggested a framework to detect malware by using the representation of the latent space using Variational Autoencoder (VAE) and a range of classic machine-learning classifiers, such as Decision Tree, Naive Bayes, Logistic Regression, Random Forest, and LightGBM. The paper has filled the research gap of high-dimensional feature dependency, low generalization to obfuscated malware, and much hyperparameter optimization of the current procedures. Results of experiments on EMBER and BODMAS datasets demonstrated that Random Forest and LightGBM obtained an accuracy of up to 99.6 with an AUC of nearly 0.999. The uniqueness is in the fact that compact latent representations can be used to reduce the dimensionality of the features and the cost of computation without performance impairment. However, the method is sensitive to the ratios of data splits and has a greater execution time on the ensemble models in large datasets.

To fill this research gap regarding the unclear comparative effectiveness of ML models in detecting malware based on API-call-based malware detection, Li et al. [14] introduced a holistic machine learning-based evaluation of the malware detection methods on the Mal-API-2019 dataset. Random Forest, XGBoost, KNN, and Neural Network models were assessed in the study with the help of TF-IDF and PCA in terms of preprocessing. The findings indicated that Random Forest and XGBoost had the highest performance with about 68 % accuracy. The new aspect of the work is its thesis-based, data-related assessment that emphasizes the importance of preprocessing and ensemble learning immensely. Nevertheless, the research is constrained to Windows API calls, and its ability to combat zero-day and fast-developing malware is lower. A hybridizing traditional machine learning and deep learning methods, Ravin et al. [15] suggested an Attention-Based Artificial Neural Network (AB-ANN) as a complete framework of malware classification. The study has addressed the weakness of poor pattern recognition and low zero-day detection ability in traditional models. The experimental findings showed that AB-ANN was highly accurate 97%, surpassing XGBoost, Random Forest, standard ANN, and Decision Tree models. The originality of the research is the use of attention mechanisms to capture malware behavior patterns better. The increased cost of computation and scalability of attention-based models, however, poses a disadvantage by restricting their applicability to real-time malware detection systems.

Chukwuani et al. [16] suggested a scalable machine learning-based real-time malware classification and threat detection system in distributed systems. The model combines a custom hybrid algorithm of the Random Forest, the SVM, the Gradient Boosting, the CNN, and the LSTM, as well as federated and online learning, to solve the gap in the research in adaptive and real-time malware detection in distributed settings. The framework was able to detect with an accuracy of more than 96% with low rates of false positives in different datasets. The novelty consists of the synthesis of hybrid ensembles, temporal behavioral learning, and privacy-preserving federated deployment. Nevertheless, the system

raises the level of complexity and computation load because of ensemble deep-learning and retraining continually. To fill the research gap of poor signature-based detection of polymorphic and zero-day malware, Kamdan et al. [17] suggested a fixed malware detection and classification model based on the Random Forest, Decision Tree, and Support Vector Machine classifiers. This research indicated that SVM had the best accuracy of 53.2 % and that the Random Forest gave balanced scores with the shortest training rate of 0.23 seconds. The originality of this work is that it analyzes the performance-efficiency trade-off of this algorithm and shows the suitability of Random Forest to analyze data statistically. Nevertheless, the use of only the static features implies comparatively low detection rates against highly obfuscated and dynamic malware.

To fill the research gap of identifying robust and computationally efficient classifiers to detect multi-class malware, Joseph et al. [18] came up with a data-driven supervised machine learning model to detect malware. In the study, the Random Forest, Logistic Regression, and the Decision Tree models were also evaluated, whereas the highest model accuracy of 96 % was recorded in the random forest model. The originality is in the fact that the comparative analysis based on accuracy, robustness, interpretability, and speed of computation in real-time applications is presented in extensive detail. Nevertheless, less complex models like Logistic Regression provided diminished performance with more complicated malware patterns than Decision Trees, which tended to overfit. Azeem et al. [19] used malware detection and classification frameworks based on machine learning, but with KNN, Extra Tree, Random Forest, Logistic Regression, Decision Tree, and neural network MLP models in order to fill the gap in research where only a few features are represented in malware detection in the IoT context. The framework used entropy-based TF-IDF features selection, feature encoding, and data balancing on the UNSW-NB15 data. The findings indicated that Random Forest obtained the greatest accuracy of 97.68%, which is the highest. The originality of the research is the feature selection based on entropy and balanced learning, which is combined to boost the detection performance. Nevertheless, the use of single data and conventional machine learning frameworks constrains the ability to apply and extend to the unknown or zero-day malware in real-world settings.[20-26]

III. METHODOLOGY

The system, developed using the Django framework, is a cyber threat intelligence-driven malware analysis platform. It collects structured threat data and processes all data using a complete preprocessing pipeline, including cleaning, normalization, and transformation, to ensure consistency and accuracy. We use feature extraction and label encoding to ensure that the training data and real-time prediction tasks use the same standardized input format. Figure 4 depicts the proposed model architecture.

A. Dataset Description

In this study, we chose to use the Microsoft Malware Prediction dataset—a large and realistic telemetry data set, offered for a Kaggle competition bearing the same name. It is based on Microsoft Defender telemetry data on Windows operating systems and is designed for use in predictive models for malware infections. It has a training set with 8.9 million machine records and a test set with 7.5 million records, each with 83 features describing various machine attributes like system information, software information, security information, hardware information, and geographical information, and so on.

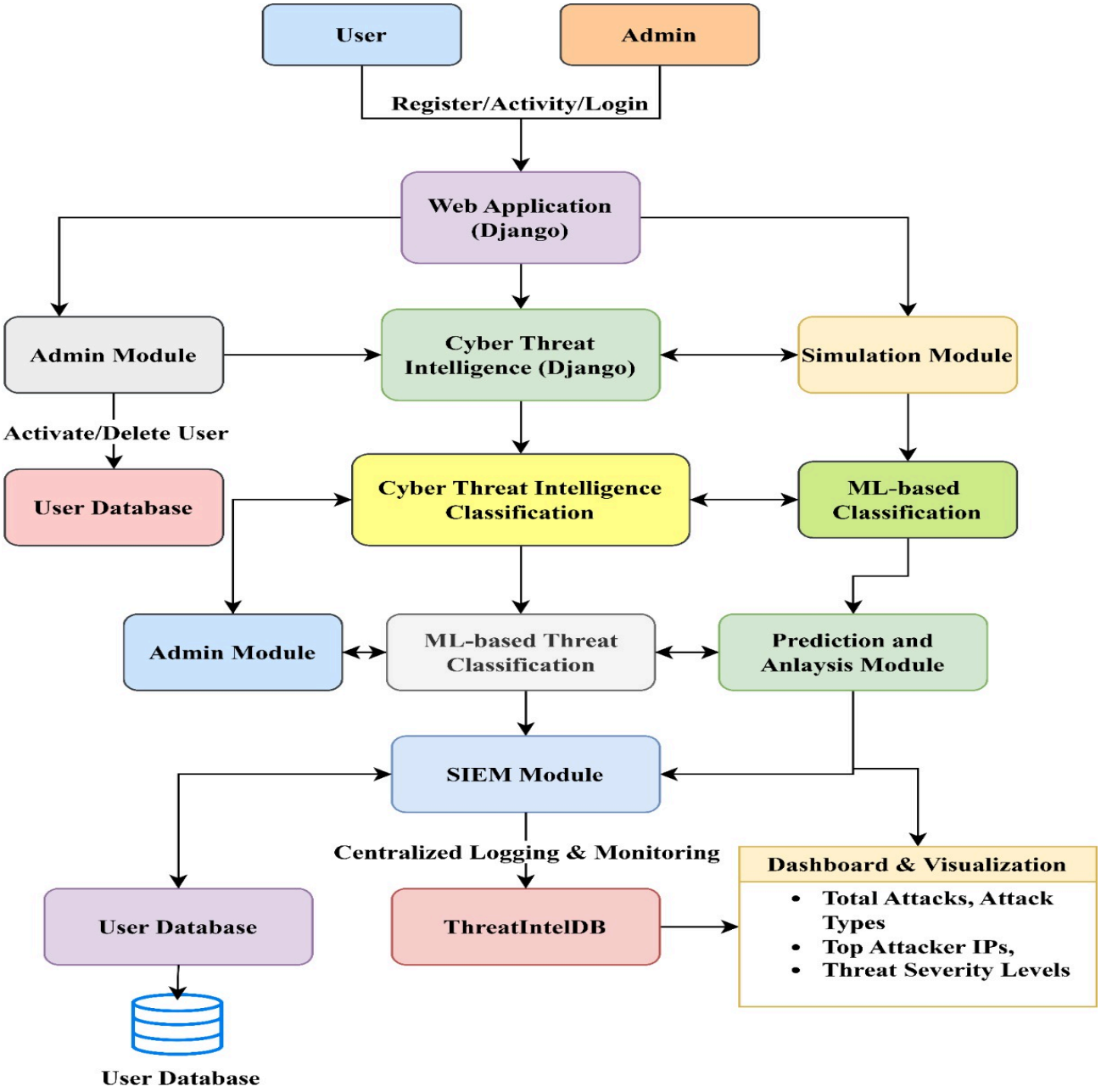

Fig 1: Graphical representation of the proposed system architecture

Essentially, it is a binary classification problem, where the target variable HasDetections tells us if malware has been detected on a particular machine or not (1 if it has, 0 if it hasn't). The predictor variables are a mix of data types, including strings for Defender version and OS version, hardware flags, census information, and so on.

B. Data Preprocessing

Before building the model, we performed extensive data preprocessing on the Microsoft Malware Prediction dataset to ensure the quality and consistency of the data and to make it machine learning ready.

Since the dataset is high-dimensional and heterogeneous in nature, we first analyzed the missing values present in each feature of the dataset. The features with a large percentage of missing values were removed to reduce noise in the dataset. Then, for the missing values present in the dataset, appropriate statistical imputation techniques were used. For numerical features, mean imputation was used:

$$x_i = \frac{1}{n}\sum_{j=1}^{n} x_j$$

where x_i represents the imputed value and n denotes the number of non-missing observations. For categorical variables, the most frequent category (mode) was used for imputation to preserve distributional characteristics.

Given the high cardinality of various properties, the dataset's predominant categorical features were converted into numerical representations using label encoding to preserve scalability. Every category c_k was associated with a distinct integer value e_k, which was written as:

$$f\ (c_k) = \ e_k \ \in Z$$

To ensure consistent feature scaling and avoid the dominance of features with larger magnitudes, numerical features were normalized. Min-max normalization was used as:

$$x_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

where, x_{min} and x_{max} indicate the feature's lowest and maximum values, respectively.

To lessen redundancy and enhance generalization, outliers and low-variance characteristics were investigated. Features with almost zero variance were also removed because they had little discriminatory value in the classification job. Malware presence is defined as follows, and the target variable HasDetections was kept as a binary class label:

$$y = \begin{cases} 1, if\ malware\ is\ detected \\ 0, otherwise \end{cases}$$

In order to maintain the original class distribution and provide a reliable and objective model evaluation, the dataset was finally divided into training and validation subsets using stratified sampling.

C. Machine Learning Classification Models

This work uses aspects of cyber threat intelligence to classify malware threats using several supervised learning techniques. Each model learns the link between the target malware label and the extracted dataset properties. Each classifier's theoretical underpinnings and mathematical formulation are explained in the ensuing subsections.

- Logistic Regression (LR): One of the most popular statistical learning techniques for binary classification issues is logistic regression. Because of its ease of use, interpretability, and efficacy on structured cybersecurity datasets, it is used as the suggested model in this study. In contrast to

linear regression, logistic regression uses a sigmoid activation function to forecast the likelihood of a class label. Considering a feature vector input:

$$X = [x_1, x_2, x_3, \dots \dots x_n]$$

A linear combination of features is calculated by the model:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

where, β_0 defines the bias term, β_i are the learned coefficients.

The logistic (sigmoid) function is used to determine the likelihood of malware presence:

$$P\ (y=1|x) = \frac{1}{1+e^{-z}}$$

A threshold, usually 0.5, is then used to make the categorization decision:

$$\hat{y} = \begin{cases} 1, & P(y=1|x) \geq 0.5 \\ 0, & P\ (y=1|x) < 0.5 \end{cases}$$

The log-loss (cross-entropy) cost function is minimized in order to train logistic regression:

$$J\ (\beta) = -\frac{1}{m}\sum_{i=1}^{m}[y_i\ log\ (\hat{y}_i) + (1-y_i)\ log(1-\hat{y}_i)]$$

This makes it appropriate for applications requiring probabilistic assurance, such as malware detection.

- Random Forest (RF): Several decision trees are combined in the RF ensemble learning process to increase classification stability and decrease overfitting. For complicated malware datasets with high-dimensional threat indicators, it works very well.

  Let, T decision trees are developed:

  $$h_1(X), h_2(X), \dots \dots, h_T(X)$$

  Every tree makes a prediction on its own. Majority voting is used to determine the final categorization output:

  $$\hat{y} = mode\{h_1(X), h_2(X), \dots \dots, h_T(X)\}$$

  RF presents randomness in two ways:

  1. Bootstrap Sampling: A randomly selected subset of the training data is used to train each tree.
  2. Random Feature Selection: Only a portion of the features are taken into account at each split.

  The impurity measures employed for splitting is often the Gini Index:

  $$Gini = 1 - \sum_{k=1}^{K} p_k^2$$

  where, P_k defines the proportion of samples to class k.

  RF is resistant against unbalanced and noisy threat data; it performs well in malware classification.

- Support Vector Machine (SVM): SVM is a discriminative classifier that creates an ideal hyperplane to distinguish between samples that are malicious and those that are not. It works best when the dataset's high-dimensional feature space has distinct class boundaries. For linearly separable data, the hyperplane is expressed as:

  $$w.\ X + b = 0$$

  where, w is the weight vector, b is the bias term.

  Maximizing the margin between the two classes is SVM's goal:

$$\min \frac{1}{2} ||w||^2$$

subject to:

$$y_i (w.X_i + b \geq 1$$

SVM uses kernel transformation for non-linear malware patterns:

$$K(X_i, X_j) = \phi(X_i).\phi(X_j)$$

Additionally, the Radial Basis Function (RBF) is a widely used kernel:

$$K(X_i, X_j) = exp(-\gamma ||X_i - X_j||^2)$$

SVM is useful in cybersecurity because it can handle subtle malware variants and complex feature spaces.

- Decision Tree (DT): A rule-based categorization model called Decision Tree divides the dataset into subgroups recursively according to feature values. It offers unambiguous decision logic for malware identification and is interpretable.

  The optimal feature split is chosen at each node by optimizing information gain:

  $$IG = Entropy\ (parent) - \sum_{j=1}^{n} \frac{|S_j|}{|S|} Entropy\ (S_j)$$

  The entropy is expressed as:

  $$\text{Entropy (S)} = - \sum_{k=1}^{K} P_k log_2(p_k)$$

  Classification is carried out at the leaf node, and the tree keeps splitting until the halting conditions are satisfied:

  $$\hat{y} = class\ (label\ at\ leaf)$$

  DTs provide quick categorization, but if they are not trimmed or mixed in ensembles, they may overfit.
- Light Gradient Boosting Machine (LGBM): An ensemble of decision trees is successively constructed by LightGBM, a sophisticated gradient boosting framework. Boosting, as contrast to Random Forest, concentrates on fixing mistakes committed by earlier trees.
  The expression for the model prediction is:

  $$\text{F(X)} = \sum_{t=1}^{T} f_t(X)$$

  where, every $f_t(X)$ define a weak DT learner.
  LightGBM uses gradient descent to minimize a loss function L at each iteration:

  $$F_t(X) = F_{t-1}(X) + \eta f_t(X)$$

  The optimization objective becomes:

  $$\text{Obj} = \sum_{i=1}^{m} L(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t)$$

  LightGBM's speed and robust predictive capabilities make it extremely effective for large-scale cyber threat intelligence datasets.

D. Deployment Architecture

A component diagram in UML represents the physical configuration of a system by depicting its major components and how they are connected. In the proposed cyber threat analysis system, the component diagram depicts how the django web application is broken down into logical pieces that include

the user interface, authentication module, machine learning processing module, and database layer. In essence, each piece is responsible for performing a certain function and communicates with other pieces using well-defined interfaces. The flow of requests from the user interface to the controller, predictions from the machine learning models, and storing and retrieving predictions are well depicted in the component diagram. Figure 2 presents the deployment architecture of the proposed system.

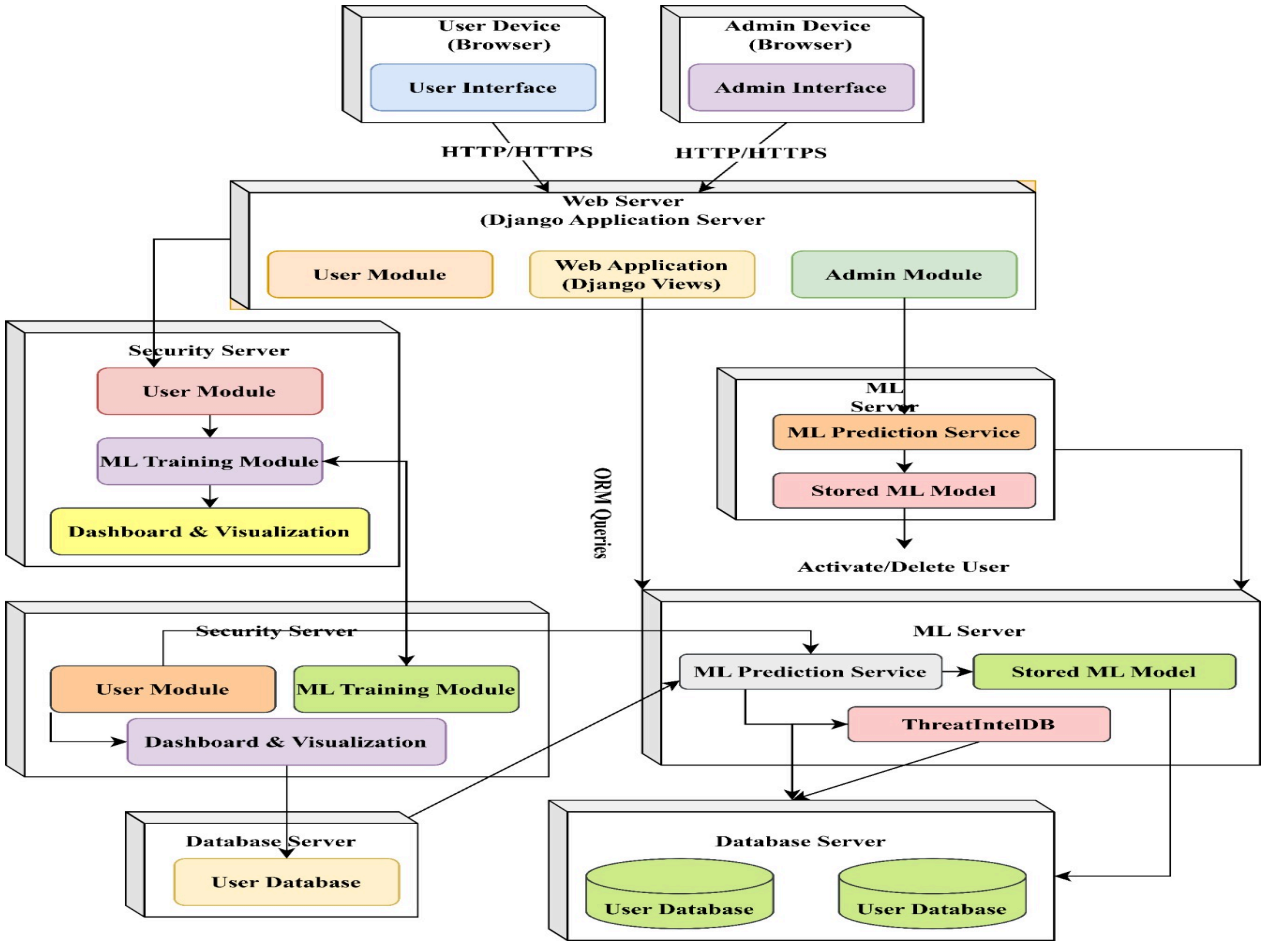

Fig. 2: Deployment architecture of the proposed system

IV. RESULTS AND DISCUSSIONS

A. ML Classifiers Performance Analysis

This is evident from the results, which clearly indicate the differences in the classifiers' performances which presented in the Table 1 and Figure 3. As seen from the findings, LR achieves an accuracy of 90.42%. This is good enough for establishing a baseline for malware detection. However, its precision and F1-score are low, indicating its inability to deal with the more complex patterns of cyber threats in the cyber threat intelligence dataset. The DT classifier manages to edge past it with an accuracy of 91.76%. This is due to its feature-based decision-making process. However, it is also more likely to overfit for high-dimensional data. Next up is the SVM classifier with an accuracy of 93.18%. This is because it is able to create the best decision boundaries for distinguishing between the malware and benign data. TheLGBM is next with an accuracy of 94.63%. This is evident from its high aptitude for dealing

with structured data with its use of iterative boosting and precise error correction. Thus, it is one of the most competitive classifiers in the list.

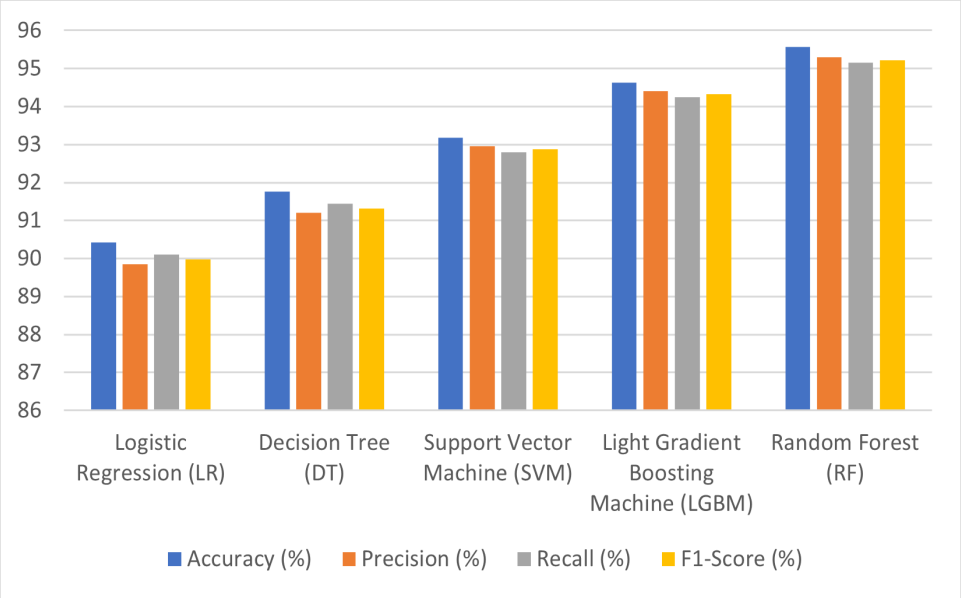

Fig. 3: Table: Performance analysis for the ML models

However, from all these classifiers, RF emerges as the best classifier with an accuracy of 95.57% and the highest precision, recall, and F1-score. This is because it is an ensemble-based classifier with multiple decision trees working together for better stability and generalization. Thus, it is evident from the output that while SVM and LGBM are good classifiers, Random Forest is the most reliable and consistent classifier for the proposed system for proactive malware threat detection.

Table 1: Performance Comparison of Assessed Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression (LR)	90.42	89.85	90.10	89.97
Decision Tree (DT)	91.76	91.20	91.45	91.32
Support Vector Machine (SVM)	93.18	92.95	92.80	92.87
Light Gradient Boosting Machine (LGBM)	94.63	94.40	94.25	94.32
Random Forest (RF)	95.57	95.30	95.15	95.22

B. ROC Curve Analysis

This is further supported by the ROC curve, which clearly demonstrates the ability of the Random Forest classifier in detecting malware for our threat detection system in Figure 4. It has an AUC of 0.96, which is an excellent measure of the separation between the two types of data: the malware and the harmless ones. The curve is closer to the top left, which is an indication of the high true positive rate and low false positive rate. This is an indication of the ability of the model in detecting malware-related

information in the cyber threat intelligence system. The high and solid AUC is an indication of the reliability and effectiveness of the classifier, which makes it the best choice for the purpose.

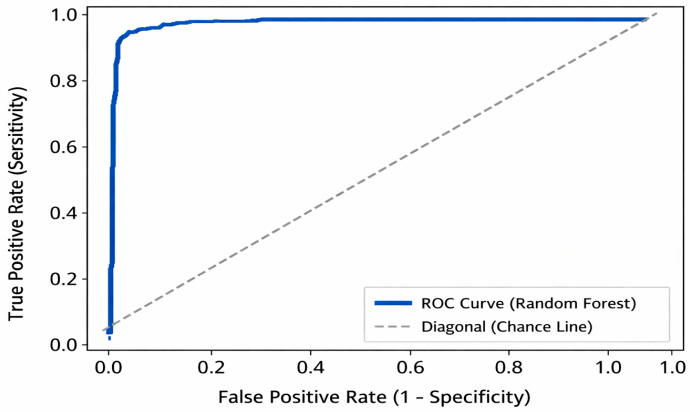

Fig. 4: ROC curve analysis for the RF model

C. Training performance Analysis

From the accuracy and loss curves of the RF model, it is clear that the model learns well and that the performance does not very much as shown in Figure 8. The accuracy curve rises smoothly over the training data, starting from 88% in the first pass to reaching the highest point at 95.57%. This shows that the model becomes better at distinguishing between malware and benignware as it goes through the training data. On the other hand, the loss curve declines smoothly from 0.45 to 0.14 over the training data. This shows that the performance of the model in terms of prediction errors reduces over the training data. The fact that the accuracy of the model improves while the loss reduces shows that the model is improving over the training data.

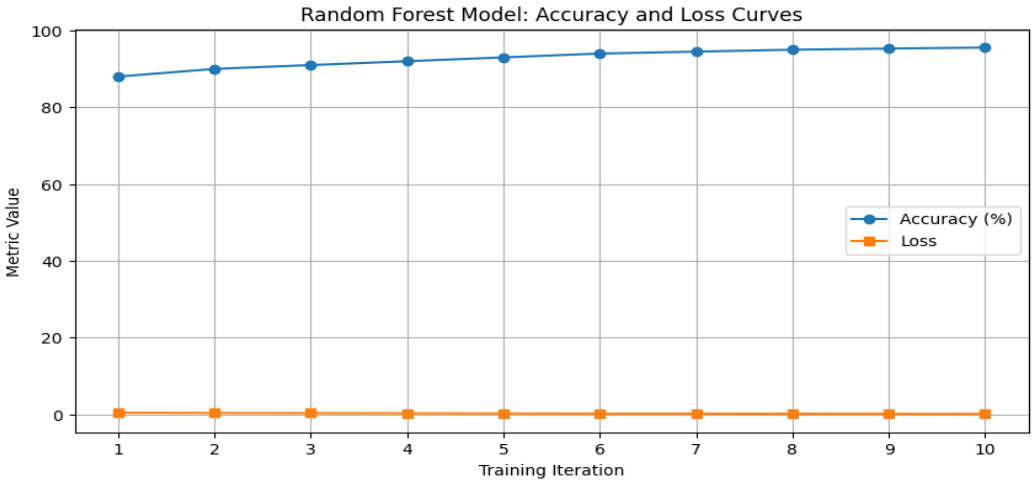

Fig. 5: Accuracy and Loss curve analysis for the RF model

D. Test Cases Analysis

The test cases include the entire functionality of the cyber threat intelligence platform, from user signup, login, profile updates, and password changes, to ensure that access control is always secure and

consistent. The test cases also include the functionality of the platform when it comes to dealing with duplicates and incorrect login credentials. To make sure that the database is updated appropriately and that sessions are always safe from unwanted access, the admin tools' ability to activate, deactivate, and delete users was also evaluated. On the machine learning aspect, the evaluation included the functionality of loading data and training models, as well as comparing performances and saving models for later use. The Random Forest classifier has the best performance, according to the evaluation, and its training characteristics have been recorded and saved for use in future predictions. Reliability and traceability were further reinforced by testing the prediction pipelines to make sure that threat classification is accurate, confidence levels are transparent, and findings are stored appropriately.

V. CONCLUSION

This study introduced a framework of analyzing the dynamic nature of malware-based cyber threats using a cyber threat intelligence-based approach. Through a critical analysis of modern malware, its propagation, evasion techniques, and actual attacks, it was demonstrated that modern malware poses a significant threat to critical infrastructure, such as healthcare, banking, and infrastructure, among others. Through the use of structured cyber threat intelligence, data preprocessing, feature engineering, and machine learning, it was shown that malicious activities could be effectively identified from large-scale high-dimensional data sets. Through a comparative performance evaluation of various machine learning models, it was shown that the Random Forest model had a higher performance compared to other models, with an accuracy of 95.07%. In addition, high precision, recall, and F1-score were achieved, demonstrating that ensemble-based machine learning models are effective in handling noisy and complex data sets, as is often encountered in machine learning-based cybersecurity models. Through its web-based system, it was demonstrated that the model is effective in classifying modern malware, as well as facilitating secure user interaction.

REFERENCES

1. Ibrahim, R. M. (2025). Enhancing multifactor authentication using machine learning techniques. *Mesopotamian Journal of CyberSecurity, 5*(2), 899–912.
2. Ramcharan, H. (2025). The effective integration of multi-factor authentication (MFA) with zero trust security. *American Journal of Mathematical and Computer Modelling, 10*(1), 1–5.
3. Asif, M., Abrar, M., Salam, A., Amin, F., Ullah, F., Shah, S., & AlSalman, H. (2025). Intelligent two-phase dual authentication framework for Internet of Medical Things. *Scientific Reports, 15*, Article 1760.
4. Tran-Truong, P. T., Pham, M. Q., Son, H. X., et al. (2025). A systematic review of multi-factor authentication in digital payment systems: NIST standards alignment and industry implementation analysis. *Journal of Systems Architecture, 162*, 103402.
5. Zeeshan, N. (2025). Continuous authentication in resource-constrained environments. *Sensors, 25*(18), Article 5711.
6. Alotaibi, A. (2025). A review of the authentication techniques for Internet of Things in smart cities. *Sensors, 25*(6), Article 1649.
7. Ganmati, A., Afdel, K., & Koutti, L. (2025). *Deep learning-based multi-factor authentication: A survey of biometric and smart card integration approaches* (arXiv Preprint No. arXiv:2510.05163). arXiv.
8. Gilbert, C., & Gilbert, M. A. (2025). Continuous user authentication on mobile devices. *International Research Journal of Advanced Engineering Science, 10*(1), 158–173.
9. Allafi, R., & Darem, A. A. (2025). Usability and security in online authentication systems. *International Journal of Advanced Applied Sciences, 12*(6), 1–12.
10. Lengert, A. (2025). *2FA: Navigating the challenges and solutions for inclusive access* (arXiv Preprint No. arXiv:2502.11737). arXiv.

11. Fahim, A., Dey, S., Absur, M. N., Siam, M. K., Huque, M. T., & Jafreen, J. G. (2025). Optimized approaches to malware detection: A study of machine learning and deep learning techniques. In *Proceedings of the 14th IEEE International Conference on Communication Systems and Network Technologies (CSNT)* (pp. 269–275). IEEE.
12. Qin, X., Li, W., & Rosenberg, P. (2025). RoundImage: Towards secure graphical password authentication via rounded image selection in IoT. *IEEE Internet of Things Journal*.
13. Mahato, A., Majumdar, R., & Ghosh, S. K. (2025). Feature-driven malware detection using cascade machine learning models. *SN Computer Science, 6*(7), Article 794.
14. Ajayi, B., Barakat, B., & McGarry, K. (2025). *Leveraging VAE-derived latent spaces for enhanced malware detection with machine learning classifiers* (arXiv Preprint No. arXiv:2503.20803). arXiv.
15. Li, Z., Zhu, H., Liu, H., Song, J., & Cheng, Q. (2024). *Comprehensive evaluation of Mal-API-2019 dataset by machine learning in malware detection* (arXiv Preprint No. arXiv:2403.02232). arXiv.
16. Suru, H. U. (2024). Improving the usability of graphical authentication systems using subject-based images.
17. Ravin, D., Akshwin, T., Thenmozhi, M., et al. (2025). Malware classification using machine learning and deep learning: A comprehensive approach. *Cureus Journal of Computer Science, 2*, 17095–17112.
18. Chukwuani, E. N., Odunsi, O. R., & Ikemefuna, C. D. (2025). Machine learning techniques for real-time malware classification and threat detection in distributed systems.
19. Dias, N. I., Kumaresan, M. S., & Rajakumari, R. S. (2023). Deep learning based graphical password authentication approach against shoulder-surfing attacks. *Multiagent and Grid Systems, 19*(1), 99–115.
20. Kamdan, Y. P., Pratama, R. S., Munzi, R. S., Mustafa, A. B., & Kharisma, I. L. (2025). Static malware detection and classification using machine learning: A random forest approach. *Engineering Proceedings, 107*(1), Article 76.
21. Joseph, H., Manjus, E., Kokatnoor, S. A., & Madavi, K. P. B. (2024). Data-driven malware detection: Exploring supervised machine learning approaches. In *Proceedings of the International Conference on Data Science, Computation and Security* (pp. 465–476).
22. Azeem, M., Khan, D., Iftikhar, S., Bawazeer, S., & Alzahrani, M. (2024). Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches. *Heliyon, 10*(1), Article e23574.
23. George, A. M., Rajan, K. T., Jambula, K. R., & Ahmed, S. T. (2025, August). Adaptive Firewall System to Predict Phishing Websites using Machine Learning Model. In *2025 International Conference on Artificial Intelligence and Machine Vision (AIMV)* (pp. 1-6). IEEE.
24. Fatima, N., Noorain, A., Ahmed, S. T., & Siddiqha, S. A. (2025, December). Automated Medical System for Rural Communities to Provide Medication without Human Interruption Using Machine Learning Techniques. In *2025 IEEE 5th International Conference on ICT in Business Industry & Government (ICTBIG)* (pp. 1-5). IEEE.
25. Alex, S., Shashank, J. T., & Ahmed, S. T. (2025, July). Machine Learning Based Network Traffic Analyser for Malicious and Benign Traffic Detection. In *2025 International Conference on Computing Technologies & Data Communication (ICCTDC)* (pp. 1-6). IEEE.
26. Ahmed, S. T., Akshaya, K. R., Vattikuti, H., Preetham, L. S. P., & Dutta, R. K. (2025, September). Dynamic Traffic Status Classification and Monitoring in Indian Metro Cities Using Edge-AI Computation. In *2025 International Conference on Vehicular Technology and Transportation Systems (ICVTTS)* (pp. 1-6). IEEE.