Vol. 07, No. 01, pp. 26 –40 (2026)
ISSN: 2708-0757

JASTT

JOURNAL OF APPLIED SCIENCE AND TECHNOLOGY TRENDS

www.jastt.org

RANSEC: Hybrid Ensemble Learning-based Secure Approach for Ransomware Detection in Cyber-Physical Defence Systems

Om Prakash Suthar[1], Mohammed Wasim Bhatt[2*], Md. Solaiman Mia[3], Gaganpreet Kaur[4], Yogeshwar Prajapati[1], Rahul Bhandari[5], Evans Asenso[6]

[1]Department of Computer Engineering, Marwadi University, Rajkot, Gujarat, India, opsuthar18@gmail.com, yogeshwarprajapati2000@gmail.com
[2]Model Institute of Engineering and Technology, Jammu, J&K, India, wasim.cse@mietjammu.in
[3] Department of CSE, Green University of Bangladesh, Dhaka, Bangladesh, solaiman@cse.green.edu.bd
[4]Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India, kaur.gaganpreet@chitkara.edu.in
[5]Department of Computer Science Engineering, Chandigarh University, Punjab, India, er.rahulbhandari1987@gmail.com
[6]Department of Agricultural Engineering, School of Engineering Sciences, University of Ghana, Accra, Ghana, easenso@ug.edu.gh

**Correspondence: wasim.cse@mietjammu.in*

Abstract

Sophisticated ransomware attacks increasingly target cyber-physical systems (CPS), therefore seriously compromising security for vital infrastructure. Stronger and more intelligent protection systems are necessary, as conventional detection systems can struggle to adapt to evolving attack patterns. This work proposes a novel hybrid ensemble learning model that is driven by artificial intelligence and makes use of weighted voting, combining Random Forest classifiers with SVM classifiers and another technique, stacking, which utilizes SVM with XGBoost as base classifiers and logistic regression as a meta classifier to improve the accuracy of ransomware detection. Experiments performed on the publicly accessible Kaggle ransomware dataset, containing 62,485 records of process, network activities, validate the superiority of the proposed approach, as the stacking-based hybrid model provides 93.15% accuracy compared to current single and ensemble classifiers. The adaptive resilience of the framework is guaranteed by the dynamic weighting, the meta-learning combination, which reduces the number of false positives and provides low-latency performance that is necessary in the real-world implementation of CPS. This secure model is the first step towards extending the existing literature and provides a scalable means to defend against future ransomware attacks on cyber-physical systems, protecting critical infrastructure in smart manufacturing, healthcare, and energy systems.

Keywords: *Cyber Physical System, Weighted Voting Mechanism, Support Vector Classifier, Artificial Intelligence, Ransomware, Extra Tree Classifier*

Received: October 03rd, 2025 / Revised: December 10th, 2025/ Accepted: January 05th, 2026 / Online: January 10th, 2026

I. INTRODUCTION

Cyber-physical systems (CPSs) integrate computer, communication, and control capabilities and are considered next-generation intelligent systems. The ability to perceive in real-time, control dynamically, and provide information service is a result of ongoing communication and deep integration of computing devices with physical processes in CPSs [1]. Cyber-physical systems are often viewed as a bridge between the physical and cyber worlds due to their calculation, communication, and control capabilities; they comprise sensors, actuators, and controllers. Multiple industries have made extensive use of them, including healthcare, smart manufacturing, smart transportation, smart grids, water supply, defence, and avionics [2], [3].

Recent years have seen the development of a variety of safety-critical systems using CPS theory and technology. These systems are vulnerable to cyberattacks since they allow communication networks to access data, services, and information about physical entities. Now that science and technology, particularly IT, are advancing at a rapid pace, cyberattacks can damage physical components. As a result, CPSs are opening themselves up to cyberattacks, which is a major concern for system security. As a result of cyberattacks' ability to alter or delete data as well as introduce misleading data, decision-makers may make poor choices. The inception of cyberattacks has the potential to trigger a cascade of failures that render CPSs inoperable and result in enormous monetary, property, and fatality losses [2].

doi: 10.38094/jastt71555

When it comes to cyber defence, ransomware assaults have recently emerged as a major concern, especially for cyber-physical systems (CPS), which combine digital controls with physical operations. Companies in the medical care, power, transportation, and production industries rely on these systems, making them straightforward targets for cybercriminals [4], [5]. Encrypted viruses that disrupt the business activities by interfering with security weak points of the networked systems can be devastating to the bottom line and the general business activities. These attacks require a higher level of advanced detection mechanisms to be put in place due to the dynamic threat environment [6].

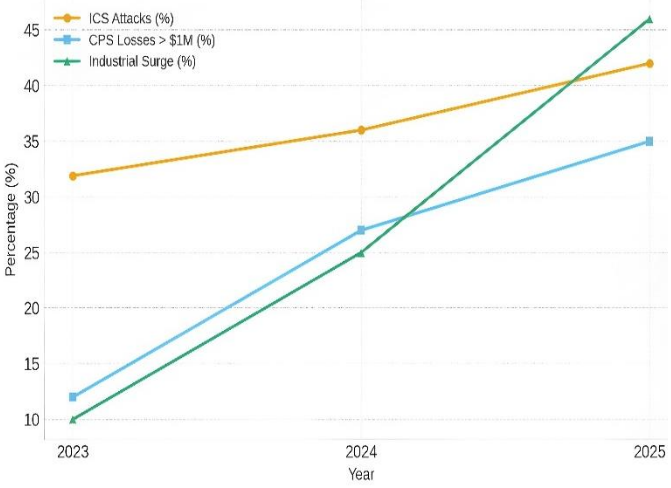


Fig. 1. Ransomware escalation trends in the CPS environment (2023-2025).

The graph in Figure 1 clearly shows the significant increase in ransomware threats in CPS/ICS settings between 2023 and 2025, where the industrial attack surge and losses of millions of dollars per annum were increasing steadily every year. It emphasizes that, by the year 2025, more than 40 percent of industrial systems would be attacked and 35 percent would suffer financial losses amounting to more than one million dollars. This fact makes the adaptive cybersecurity solutions urgent, which puts the research gap of the given research, that is, the development of a hybrid ensemble structure to detect the real-time CPS ransomware with a measurable effect on the infrastructure protection.

It has brought a revolution in dealing with cyberattacks using AI. The solutions provided by this technology are data-driven and scalable, and contribute to better detection and response time. Ensemble learning is an example of the most promising AI techniques that have been established based on the capability of numerous models to make joint predictions [7]. To identify ransomware, this research study suggests a novel hybrid ensemble learning framework that is both a stacking and a weighted voting. Weighted voting combines model performance, and stacking is a type of meta-model in order to gain knowledge about forecasts of base models. This is the reason why the system is stronger and more flexible in the final analysis [8].

These ensemble techniques can be combined to help recognize various variants of ransomware, improve the timely decision-making process, and reduce the number of false alarms. This paper emphasizes the applicability of the approaches that imply the implementation of AI to protect CPS against cyberattacks, to guarantee that it will not affect the regular working of the system and render it resilient. A fast, adaptable, and accurate answer to the acute problem with ransomware in the context of cyber-physical environments is provided, which adds to the delivery of the relatively new field of cybersecurity [9].

Despite significant advancements, current ransomware detection models of CPS do not usually have real-time flexibility, do not reduce false positives, or cannot be extended to fully scale to real-life and complicated deployments. The proposed research will close this gap in critical research by creating a dual-layer hybrid ensemble model that has been designed with the specific aim of detecting ransomware in cyber-physical systems in a manner that is both secure and efficient.

A. *Motivation*

The increasing cases of ransomware attacks on cyber-physical systems (CPS) [10] present great requirements of precision, versatility, and speed [11-13]. Conventional ensemble techniques (e.g., bagging, boosting) have proven to be ineffective to some degree, but they are not able to handle heterogeneous, non-stationary data and evolving threat patterns common in CPS settings. According to recent research, the hybrid method, which consists of combining several structurally different models and meta-learning, is better than simple ensembles in terms of generalization and error correction. Thus, the objective of the work is to create a dual-layer hybrid ensemble, quantitatively evaluate its benefits and optimize with low-latency, real-world CPS ransomware protection.

B. *Novelty*

The novelty of the study has explicitly entailed a structured, systematic comparison of two advanced paradigms of hybrid ensembles, namely dynamic weighted voting and stacking-based meta-learning as applied to the ransomware detection in CPS environments. This study presents a dual-layered hybrid architecture and officially compares the behaviour of both strategies in terms of the functioning, the nature of decision fusion, the delay of inference, the stability of generalization, and the calibration of probabilities in both ensemble strategies under the same experimental setting.

The strategies that the current study takes into account are both the approaches adopted by other researchers in the past: (i) wherein they only adopt a single ensemble mechanism, a (ii) dynamic weighted voting scheme which dynamically adjusts the contribution of classifiers based on real-time measures of performance (precision, recall, and F1-score), and (iii) a meta-learning layer and logistic regression as a strategy to eliminate errors generated by various base learners (SVM, RF and XGBoost). Not only they become stronger against the changing multimorphic ransomware forms, but they also reduce overfitting and generalization as a result of such synergy.

The system is evaluated on large-scale, rich-featured data from 62,485 ransomware and benign cases, which is far broader than most existing research, which uses tiny datasets or fictional data. Other distinctive features include our real-time adaptation system, which provides CPS without disturbing operations due to a low-latency design and real-time performance. Finally, we find that our hybrid using stacking has a cross-validation accuracy of 93.00%, largely due to a balanced precision-recall trade-off with lower log-loss than conventional ensemble and standalone classifiers and offers resilience and reliability to mission-critical systems. AUC of 0.96 shows that the ensemble can distinguish ransomware from non-malicious occurrences at both ends of the categorization range.

C. *Contributions*

The main contributions in this study are as follows:

- **Dual-Layer Hybrid Ensemble Design**— This paper presents a new ensemble design, the combination of dynamic weighted voting and stacking meta-learning models to deal with the specific problems that CPS faces, like data diversity, real-time adaptability, and the ability to withstand sophisticated hazards (refer to Section 3 for framework design).
- **Evaluation on large-scale realistic datasets**—On a large-scale dataset of 62,485 ransomware and benign examples, including system calls, file changes, and network traffic, our model is tested (refer to Section 4 for evaluation practice). The fact that we have a large and rich dataset guarantees that our results can be scaled, are representative, and are more reflective of real-world CPS deployment situations than other studies that used smaller or synthetic datasets.
- **Real-Time Adaptability to CPS**— The proposed structure is optimized for low latency and simultaneous processing to ensure ransomware detection and mitigation without affecting CPS activities. This dynamic is crucial in the most important structures, where false positives and detection delays can be disastrous.
- **Superior Comparative Performance**— In experiments, our stacking-based hybrid ensemble achieved 93.00% cross-validation accuracy with similar precision-recall and lower log-loss, outperforming individual classifiers (SVM at 84.95%, RF at 93.15%) and other ensemble baselines. These results demonstrate the framework's accuracy and generality, which are ideal for mission-critical CPS applications (refer to Section 5 for comparative results and discussion).

D. *Paper Organization*

The subsequent portions of the article are managed as follows. Section 2 surveys the relevant literature on AI-driven ransomware detection. Section 3 elucidates the functionality of our suggested hybrid models. Section 4 delineates the assessment criteria for the suggested hybrid methodologies. Section 5 evaluates the results of these hybrid methodologies and contrasts them with the established approaches. Section 6 ultimately closes our analysis and delineates future possibilities regarding ransomware assaults.

II. Related Work

This section explores the literature review that is relevant to the proposed work and includes the theoretical framework and original contributions by the field of artificial intelligence and its subfields. The key considerations of the studies have been presented in Tables I and II.

A. *Seminal Contribution*

Guan Li et al. (2024) have presented a model that would combine the predictive capabilities of DL models with the ability of Monte Carlo Tree Search (MCTS) to locate a multifaceted solution to different kinds of ransomware. The accuracy, as well as low false positives, demonstrated that the hybrid framework had superior performance, equivalent to the traditional ML models, since it was thoroughly tested. The system was enriched with MCTS, which provided the possibility to examine alternative decision paths in case of the emerging threats because of the real-time response capability of the system. The proposed paradigm was also efficient in computing, and this made it well-suited for real-time implementation at the business level.

A nimble and effective method of mitigation of ransomware threats, the results reveal that the hybrid system has the capacity of a formidable defence system in contemporary cybersecurity [14]. Stephen Venne et al. (2024) used the Pattern-Entropy Segmentation Analysis (PESA) framework in their study, whereby detection of ransomware can be done at a more specific and faster rate with the assistance of entropy analysis of network traffic in real-time. PESA is also based on entropy changes to identify early indicators of maliciousness generated by the ransomware process of encrypting files, rather than the common signature or behavior-based methods, to avert severe damage before it can occur. They test it on a simulated environment of a network and prove that it can recognize a variety of ransomware strains at a high rate with few false positives and a quick reaction time. Furthermore, the system is resistant to obfuscation; therefore, it is an authoritative contender to the applications of cybersecurity in real-life scenarios. To improve the security of the network and minimize the damage of ransomware infections, the findings suggest the possibility of the practical importance of entropy-based detection [15].

A new method is offered in the article by Samuel Wasoye et al. (2024), and it is based on the principle of applying machine learning models that employ a BTLS (Binary Transformation and Lightweight Signature) algorithm to make ransomware detection faster and more accurate. The fact that it is possible to extract the static and dynamic information of ransomware samples created by the BTLS algorithm allows us to analyse and classify them more deeply. As the experimental results have shown, both sets of features combined increased the accuracy of classification greatly, and the fact that the algorithm is designed to minimize the rate of false positives makes it suitable to be used in the real world.

In order to overcome failures of conventional forms of detection solutions and create machine learning-based cybersecurity solutions, the suggested solution will provide a scalable approach that will be able to adjust to the appearance of new forms of ransomware [16]. In this research, Jiugang Chen et al. (2024) take into account the degree of machine learning algorithm recognition of attack on the decentralized storage system known as the Interplanetary File System (IPFS) based on ransomware. The experiment gauges the correctness, accuracy, remembrance, and strength of the various ML models in an unfavourable environment, which are DT, LR, RF, GBM, and CNN. The results show that further advanced models like random forests (RF), convolutional neural networks are more powerful, more accurate and less evasive. The results highlight why enhancing IPFS concerning the counteraction of ransomware attacks is possible within the possibilities of implementing machine learning in the cybersecurity practices of decentralized networks. This should be followed by future work to make the models more suitable to emerging threats, more varied in terms of datasets, and their usability in more operational scenarios [17].

The cross-validation strategy employed by SUBIR PANJA et al. (2025) is 5-fold, in accordance with the dataset it gathered to study it. They estimate the memory and execution time demands of each of the 14 iterations of the ML models, both with the complete feature set and with the subset of the features that have survived the data preprocessing stage. They applied the Extra Tree classifier (ETC) to detect the top ten important characteristics by Gini impurity scores to achieve more accuracy and reduce the time needed to arrive at the results. Thereafter, they analyzed the experimental findings and discovered that the RF classification model, when applied to the set of reduced features, achieved ROC-AUC scores of 0.99 and an accuracy for prediction of 99.39%. Results that are consistent with the ETC model prediction prove that the proposed model can work. A very modest standard deviation indicates that the suggested model is robust. In addition to being very responsive, it has a low memory usage and execution time [18].

Juan A. Herrera-Silva et al. (2023) have developed a method that can identify both existing and future forms of this hazard. Listed below are the goals of this study:(1) Use a sandbox to test out variations of encryptor and locker ransomware with goodware, to create JSON files that include dynamic settings. (2) Determine which dynamic features are most useful for distinguishing encryptor and locker ransomware from legitimate software, and then pick the least redundant ones. Using these chosen parameters for examples of various artefacts, develop and make public a dynamic attributes dataset. Utilize the dynamic feature dataset alongside machine learning methods to create models. Over the course of the evaluation, 20 types of ransomware and 20 types of goodware were examined across five different platforms. Every one of the 2000 entries in the final feature collection has 50 attributes. This dataset enables an ML detection using a 10-fold cross-validation, with neural networks, random forests, and XG Boost boosted regression trees all achieving average accuracy superior to 0.99 [19].

Amjad Alraizza et al. (2023) proposed research to examine the present state of automated ransomware detection and to speculate on its possible future debates. This document offers a detailed description of ransomware, a chronology of attacks, and background information. Moreover, it provides in-depth research on the existing strategies of ransomware mitigation, prevention, and recovery. This research has additional benefits, such as an analysis of studies conducted between 2017 and 2022. Here, readers can obtain the latest information about ransomware detection methods and how they have evolved to fight these assaults. This study concludes that there are still many questions about ransomware detection and several possible obstacles to further research in this area [20]. Robert Bold et al. (2022) present a comprehensive literature review on ransomware detection with advanced ML models. The outcomes, however, indicated that previous attempts often emphasized accuracy while neglecting the importance of other values in the confusion matrix, such as false negatives. Hence, they have utilized a dataset containing 730 malicious and 735 benign samples to assess the efficacy of ML models in mitigating ransomware at various points in a detection system's design, as well as the associated costs. The results demonstrate that an ANN model is optimal due to its 98.65% accuracy, 0.94 Youden's index, and 76.27% net benefit; however, the RF model, with a minimum accuracy of 92.73%, offered the advantage of a 0.00% false-negative rate. The predictable cost of resources required to filter false-positives contrasts with the risk of a false-negative in this system, which resembles the unpredictable but frequently significant cost associated with ransomware infection [21].

Mohammad Masum et al. (2022) proposed an approach for ransomware identification and mitigation that relies on feature selection and uses several ML techniques, including neural network-based designs, to classify security levels. To classify ransomware, they used a variety of ML methods, including DT, RF, NB, LR, and classifiers based on Neural Networks (NN). In order to test their methodology, they have only used one ransomware dataset. In comparison to other approaches, RF classifiers achieve higher accuracy, F-beta, and precision scores, as shown in the experimental findings [22]. DARYLE SMITH et al. (2022) have discussed a ransomware detection approach and the ML methods that are commonly used to identify and understand these malicious programs and their dynamic traits. A comprehensive evaluation of those frameworks is also something those involved in cybersecurity will get from this research. Further details, such as the datasets used and the difficulties each framework may encounter when accurately recognizing different types of ransomware, will be added to this. Overall, this report provides a comparative analysis that can serve as a reference point for other colleagues in detecting ransomware [23].

B. *Key Considerations*

The most significant feature would be associated with the evaluation of the comparison of a set of ransomware detection techniques, emphasis on the features they implement, the algorithms they use, data sets, how they preprocess data, and what are their main contributions. It outlines the various approaches that consist of deep learning models (CNN, ResNet-50) and machine learning based classification (RF, SVM, XGBoost). This comparison shows the pros and cons of current approaches, resulting in Table I's ransomware detection trends for Cyber-Physical Systems (CPS).

The review shows that most current efforts use standard or single-layer ensemble techniques, which may be insufficient for CPS-specific challenges and scalability. The research proposes an efficient, diverse dual-layer hybrid ensemble for real-time heterogeneous CPS to cover these shortcomings.

TABLE I. COMPARISON OF RANSOMWARE DETECTION APPROACHES

Authors	Features	Algorithm	Dataset	Preprocessing	Key Contribution	Limitations
J. E. Hill et al. (2024) [24]	HPC, MCC	Ensemble, SVM, KNN, NN	Real-world datasets (20 benign, 15 Ransomware)	Normalization Data Cleaning,	Optimizing hardware performance counters	Limited samples, system dependencies
B. Keyogeg et al. (2024) [25]	File access patterns, Process creation anomalies, N/W traffic behaviors	RF, SVMs	Log datasets	Feature Extraction, Normalization	Simulated Active Directory environment	Higher FP rates, Limited ransomware samples
D. Gihavo et al. (2024) [26]	File access patterns, Temporal modification patterns	RF, SVMs, NN	Synthetic	Log data recording, Feature extraction	Synthetic dataset generation for realistic simulation	High FP rates, Feature Noise
J. Kirkland et al. (2024) [27]	Entropy values, Modification timestamps,	RF, SVM	Custom dataset (Virus Share, Hybrid Analysis)	Entropy Calculation, Feature Extraction	High performance across file types	Limited dataset, False negatives
Y. C. Wu et al. (2024) [28]	File access, System calls, Encryption routines, Anomalies	RF	Benign Files (Linux), Ransomware	Normalization, PCA, Imputation	Hyperparameter tuning, Class balance handling	Overfitting Unknown ransomware detection issues
R. Bold et al. (2022) [21]	API Call Frequency, Process ID, Function Count	SVM, KNN, DT, RF, LR	Crypto Ransom	StandardScaler	SMOTE for class balancing, RF for accuracy	Overfitting, Limited ransomware families

TABLE II. COMPARISON OF RANSOMWARE DETECTION APPROACHES

Authors/Year	Approach Applied	Real-Time	Low False Positives?	Accuracy (%)	F1-Score	Deployment Ready?
J. E. Hill et al. (2024) [24]	ML	Yes	Yes	95%	—	No
B. Keyogeg et al. (2024) [25]	ML	No	No	94.2%	0.884	No
D. Gihavo et al. (2024) [26]	ML (RL)	Yes	Yes	93.0.6%	0.94	Yes
J. Kirkland et al. (2024) [27]	ML	No	Yes	95.6%	0.951	No
Y. C. Wu et al. (2024) [28]	ML	No	Yes	94%	0.94	No
R. Bold et al. 2022 [21]	ML	No	Yes	95.9%	0.962	Yes

III. HYBRID ENSEMBLE LEARNING BASED CYBER-PHYSICAL DEFENCE SYSTEMS

In this section, we propose a hybrid ensemble learning model to enhance the accuracy of ransomware detection. This model makes use of weighted voting to combine RF and SVM classifiers, as well as another technique called stacking that uses SVM with XGBoost as base classifiers and logistic regression as a meta-classifier. This section consists of a proposed system overview and the flow of the proposed method.

A. Proposed System Overview

The proposed framework in fig. 1 presents a cybersecurity design for Cyber-Physical Systems (CPS) that targets servers in the cloud, IoT devices, & enterprise networks, effectively addressing significant cyber threats such as phishing, malware, threat actors, and zero-day exploits. The system also incorporates an extensive system threat monitoring and a reporting system that inspects system event logs of such platforms as Linux, Windows, and other operating systems and industrial control systems (ICS), network logs, and file operations. Data obtained is processed by feature extraction and preprocessing, detection of anomalies, feature selection, and normalization to enhance a better representation of threats. The ransomware attacks are classified using a hybrid model of classification depending on machine learning that incorporates the RF, SVM, & XGBoost and assessing the system performance with the measures of precision, accuracy, and log-loss. This methodology aims to enhance the cybersecurity resilience of CPS setups and improve their threat detection efficacy.

Implementation Environment and details: The whole framework was built in Python 3.10, and the scikit-learn and TensorFlow libraries were used to build, train, and test the models. 100 estimators for random forest with max_depth = 10 and a linear kernel function for SVM were used. The stacking meta-classifier used logistic regression with a penalty of "l2" and a C value of 0.1. As shown in Figure 2, weighted voting and stacking can be combined by first aggregating base classifiers (RF, SVM) before their values are input into the meta-classifier. Every block and arrow in the diagram corresponds to a single transformation step of the data, as stated in Section 3. We used an Intel Xeon E5-2680 v4 (2.4 GHz, 16 cores) system having 128 GB of RAM and 1 NVIDIA RTX 3090 GPU to run experiments. The average time to train a model was about 90 minutes for each cross-validation fold.

B. Method Flow of Hybrid models

1) Hybrid Model using a weighted voting mechanism:

Given a classification problem with a dataset of n samples with associated labels. We aim to combine the predictions of two base classifiers, SVM and Random Forest, to produce a more accurate final prediction. This is done using a weighted voting mechanism.

Methods Used as Foundational Models: SVM and RF

1. Support Vector Machine (SVM):

One supervised learning technique that seeks to optimize the gap between two classes is SVM. The model is defined by the optimization problem using equation (1):

$$\min_{w} \frac{1}{2} ||w||^2 + C\sum_{i=1}^{n}(\epsilon_i) \quad s.t.\ y_i(w, x_i + b) \geq 1, \forall i = 1, 2, \ldots n. \qquad (1)$$

where,

- w is the weight vector normal to the decision boundary.
- b is the bias term.
- C is a regularization parameter > 0
- ϵ_i is a slack variable for misclassification
- x_i is the feature vector for the i-th sample.
- y_i is the true label for the i-th sample, where $y_i \epsilon \{-1, 1\}$.

SVM produces a decision boundary, and for a given test point x_j, it assigns a label $\hat{y}_j^{SVM}$ based on the decision function as mentioned in equation (2):

$$\hat{y}_j^{SVM} = sign(w \times x_j + b) \qquad (2)$$

Moreover, SVM can also provide probabilistic outputs (using Platt scaling) for each class, which are the probabilities mentioned in equation (3):

$p^{SVM}(y_j = 1|x_j)$, for class 1, and

$$p^{SVM}(y_j = 0|x_j) = 1 - p^{SVM}(y_j = 1|x_j) \qquad (3)$$

2. Random Forest (RF):

As a kind of ensemble learning, Random Forest employs a number of decision trees to carry out categorization tasks. In order to make a final prediction, all of the trees are trained using a different selection of features and samples. According to equation (4), let the predictions of the *k-th* tree for the test sample. x_j be symbolized by $\hat{y}_j^{RF(k)}$. The predicted label for Random Forest is the majority vote from all trees:

$$\hat{y}_j^{RF} = mode(\hat{y}_j^{RF(1)}, \hat{y}_j^{RF(2)}, \ldots\ldots, \hat{y}_j^{RF(T)}). \qquad (4)$$

Where T is the total number of trees in the forest.

Random Forest can also provide probabilities for each class by averaging the probabilities of all trees using equation (5). where,

$$p^{RF}(y_j = 1|x_j) = \frac{1}{T}\sum_{k=1}^{T} p^{RF(k)}(y_j = 1|x_j) \qquad (5)$$

and similarly for class 0.

Weighted Voting Mechanism: The weighted voting process assigns a weight to each model's predictions according to their correctness on the validation data set. These weights reflect the relative confidence of each classifier in its predictions.

Weights Calculation: The weight assigned to each classifier is proportional to its accuracy on the test set. Let the accuracy of SVM and RF on the test set be denoted by equations (6) and (7):

$$Accuracy_{SVM} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(\hat{y}_j^{SVM} = y_j) \qquad (6)$$

$$Accuracy_{RF} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(\hat{y}_j^{RF} = y_j) \qquad (7)$$

Where $\mathbb{I}\odot$ is the indicator function.

The weight for $SVM(w_{SVM})$ and $RF(w_{RF})$ Are calculated using equations (8) and (9) as follows:

$$W_{SVM} = \frac{Accuracy_{SVM}}{Accuracy_{SVM} + Accuracy_{RF}} \qquad (8)$$

$$W_{RF} = \frac{Accuracy_{RF}}{Accuracy_{SVM} + Accuracy_{RF}} \qquad (9)$$

These weights ensure that the classifier with better performance on the test set contributes more to the final prediction.

Weighted Probability Combination: The final prediction for each test sample x_j It is computed by combining the probabilistic outputs of both models using their respective weights. By using Equation (3) and Equation (5), the weighted probabilities are calculated using Equation (10):

$$weighted\ Probs_j = W_{SVM} \cdot p^{SVM}(y_j = 1|x_j) + W_{RF} \cdot p^{RF}(y_j = 1|x_j) \qquad (10)$$

where,

➢ $p^{SVM}(y_j = 1|x_j)$ is the probability that *SVM* assigns to class *1* for x_j.

- $p^{RF}(y_j = 1|x_j)$, the probability that RF assigns to class (x_j). Likewise, the revised equation (10) of class 0 can be expressed as equation (11):

$$weighted\ Probs_j^{(0)} = W_{SVM} . (1 - (p^{SVM}(y_j = x_j)) + W_{RF} . (1 - (p^{RF}(y_j = 1|x_j)) \quad (11)$$

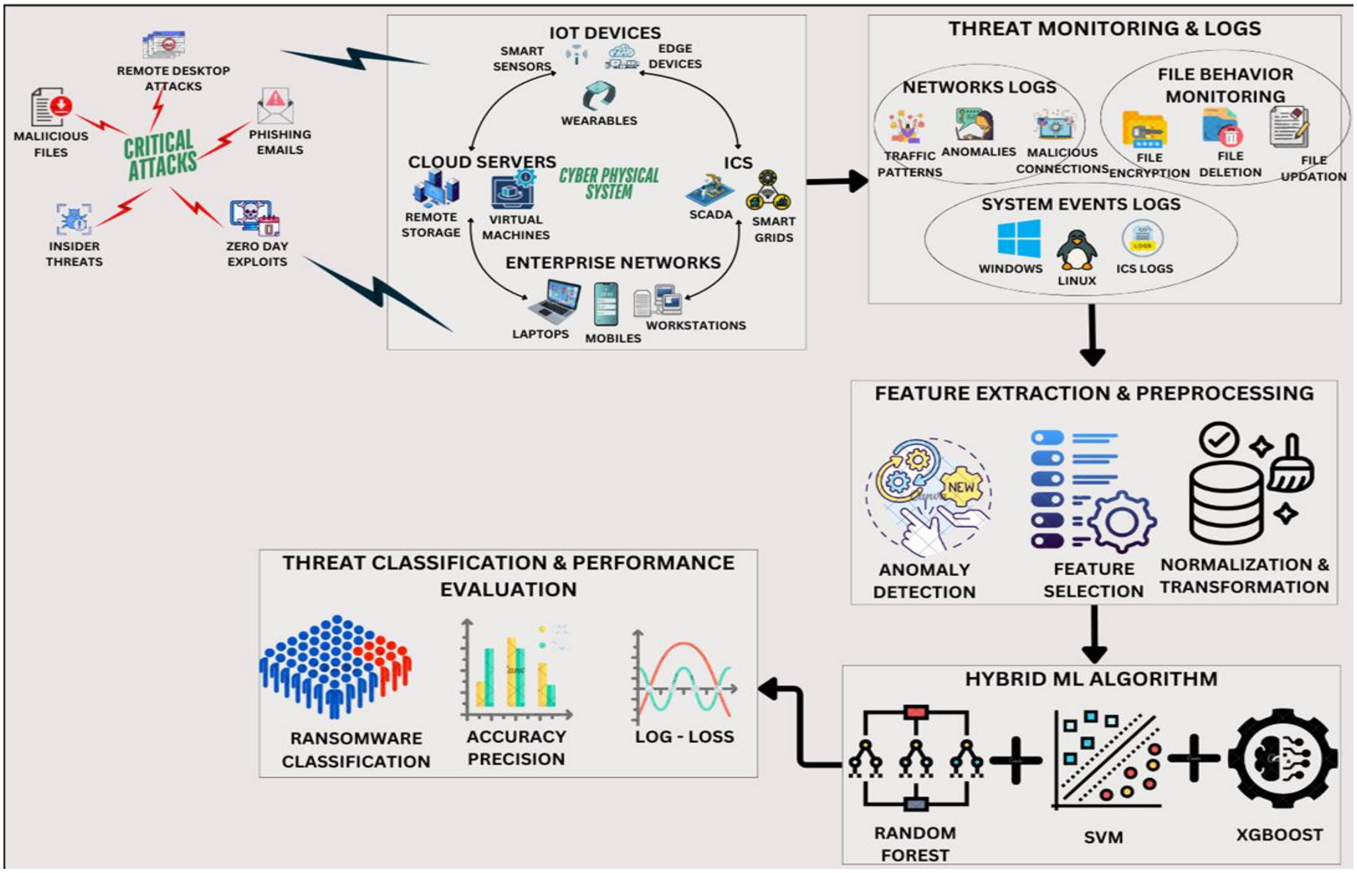
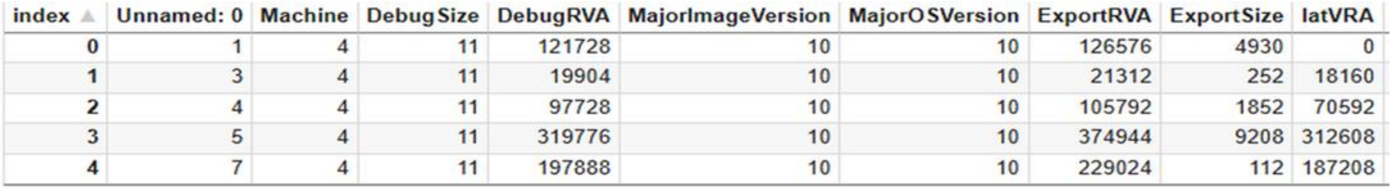
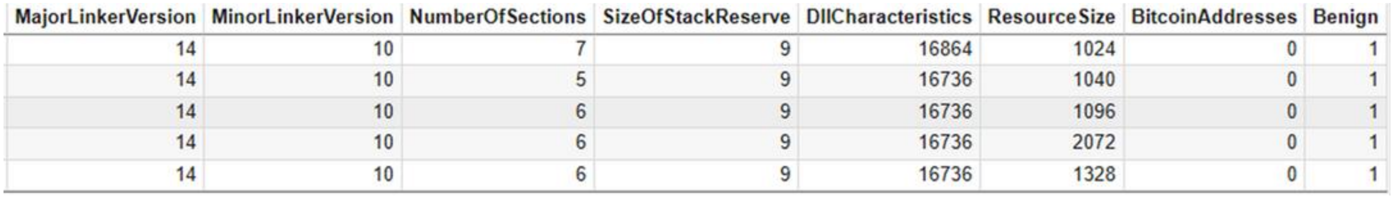


Fig. 2. Overview of Cyber Physical System (CPS) Environment.



index ▲	Unnamed: 0	Machine	DebugSize	DebugRVA	MajorImageVersion	MajorOSVersion	ExportRVA	ExportSize	IatVRA
0	1	4	11	121728	10	10	126576	4930	0
1	3	4	11	19904	10	10	21312	252	18160
2	4	4	11	97728	10	10	105792	1852	70592
3	5	4	11	319776	10	10	374944	9208	312608
4	7	4	11	197888	10	10	229024	112	187208

Fig. 3. Ransomware detection dataset with 8 features.



MajorLinkerVersion	MinorLinkerVersion	NumberOfSections	SizeOfStackReserve	DllCharacteristics	ResourceSize	BitcoinAddresses	Benign
14	10	7	9	16864	1024	0	1
14	10	5	9	16736	1040	0	1
14	10	6	9	16736	1096	0	1
14	10	6	9	16736	2072	0	1
14	10	6	9	16736	1328	0	1

Fig. 4. Ransomware detection dataset with another 8 features.

Equation (12) shows the final prediction for the test sample. x_j. is made by selecting the class with the highest weighted probability using equations (10) and (11) as follows:

$$\hat{y}_j^{final} = arg\ max(Weighted\ Probs_j, Weighted\ Probs_j^{(0)}) \quad (12)$$

LR-based Hybrid Model with Stacking Classifier: This is a hybrid model that works well in combining several classifiers to enhance predictive accuracy and strong classification. With the help of both ensemble and linear classifiers, it proves to be a complicated method of solving complex machine learning issues, which is why it can be applied in different circumstances, such as in cybersecurity and ransomware detection, as implied by the circumstances of the dataset employed.

Base Classifiers:

1) **XGBoost Classifier:** XGBoost (Extreme Gradient Boosting) is a strong ensemble approach to learning that employs the gradient boosting techniques. It is specifically efficient with structured data and is efficient when dealing with a high-dimensional data set.

2) **SVM_Classifier:** This is an extension of the SVM algorithm [29], which does the same but in the feature space, i.e., in the space of features. A linear kernel is employed in this case, and it applies to linearly separable data.

Meta-Classifier:

1) LR: Logistic regression is the last estimator, where the predictions of the base classifier are used to arrive at the final prediction. It is an effortless but efficient technique of binary classification.

Stacking Mechanism:

The Stacking Classifier is a combination of the XGBoost and SVM predictions. During training, it learns how to put these predictions together in the best way possible to get the most accurate results. We use cross-validation (CV=5) to ensure that the predictive algorithm can be generalized well by testing how well it works with other, different parts of the training data. Stacking classifiers proficiently aims to improve accuracy by using a wide range of base models, optimizing predictions through meta-learning, and employing cross-validation. This method is stronger and less prone to overfitting because it uses the learning of several classifiers. This kind of ensemble strategy is very helpful for complex classification problems where one model can't handle all the details of the data.

A StackingClassifier that uses both XGBClassifier and SVM_Classifier has a number of advantages that make the model work better. The blended approach utilizes the best of both algorithms. XGBClassifier is good at finding intricate patterns and interactions in data because of its gradient boosting structure, which makes it resistant to overfitting. SVM_Classifier is good at setting up robust classification constraints in highly dimensional environments, especially when the data can be separated linearly. Such heterogeneity gives the stacking model the advantage of the various learning strategies, enhancing the generalization to unknown data. Also, the meta-classifier may be trained to correct the mistakes of the base classifiers, which again increases the accuracy by repairing the mistakes. Cross-validation is used in the training so that the model is tuned and the likelihood of overfitting is reduced, thereby resulting in a more reliable and accurate predictive model in general.

Mathematical Model for Stacking Classifier Using XGBoost, SVM, and Logistic Regression: Suppose a set of data labelled by equation (13):

$$D = (x_i, y_i)_{i=1}^{n} \tag{13}$$

where each $x_i \epsilon R^d$ represents a feature vector and $y_i \in$ {0, 1} denotes the binary class label. Our target is to build a classification model f(x) that guesses the label. ŷ for a new, unseen sample x by merging the predictions of multiple base models via a meta-classifier.

In the first step, we create a base model using XGBoost and SVM. XGBoost is a tree-based boosting technique that builds a sequence of DTs, where each tree adjusts errors made by the previous trees. For a given feature vector x_i, the output of the XGBoost model is the probability of the positive class mentioned in equation (14):

$$p^{XGB}(y = 1|x_i) = \sigma\left(\sum_{k=1}^{T} f_k(x_i)\right) \tag{14}$$

where $f_k(x_i)$ is the output of the k-th DTree.

T is the total quantity of trees and $\sigma(z)$ is defined as: $\sigma(z) = \frac{1}{1+e^{-z}}$.

$$\hat{y}_i^{XGB} = \begin{cases} 1, & p^{XGB}(y = 1|x_i) > 0.5, \\ 0, & otherwise \end{cases}$$

SVM identifies a hyperplane in the space of features that optimizes the margin between the two categories. For a linear kernel, using Equation (2), the decision function is:

$$h^{SVM}(x_i) = W\ .x_i + b$$

where w is the weight vector, and b is the bias term. The probabilistic output of the SVM model is computed using Platt scaling as mentioned in equation (15):

$$p^{SVM}(y = 1|x_i) = \frac{1}{1+e^{-h^{SVM}(x_i)}} \tag{15}$$

The predicted label for SVM is mentioned using equation (16):

$$\hat{y}_i^{SVM} = \begin{cases} 1, & h^{SVM}(x_i) > 0, \\ 0, & otherwise \end{cases} \tag{16}$$

In the second step, stacking combines the predictions from base models using a meta-classifier. In our case, we use logistic regression as the meta-classifier. The base models (XGBoost and SVM) are first trained on the training data. (X_{train}, y_{train}). For each sample x_i in the dataset, using Equations (14) and (15), the base models generate predictions in the form of probabilities as per Equation (17):

$$P_{base}(x_i) = \begin{matrix} p^{XGB}(y = 1|x_i) \\ p^{SVM}(y = 1|x_i) \end{matrix} \tag{17}$$

Thus, by using equation (18), the transformed dataset for the meta-classifier is:

$$Z = \{(P_{base}(x_i), y_i)\}_{i=1}^{N} \tag{18}$$

In the third step, the meta-classifier takes the output probabilities $\boldsymbol{P_{base}}(x_i)$ as input features and learns to predict the final class label y_i LR models the probability of the positive class mentioned in equation (19) as:

$$p^{Meta}(y=1|P_{base}(x_i)) = \sigma(w_{meta}.P_{base}(x_i) + b_{meta}) \quad (19)$$

The final predicted label is calculated using Equation (20) as:

$$\hat{y}_i^{final} = \begin{cases} 1, & p^{Meta}(y=1|P_{base}(x_i)) > 0.5, \\ 0, & otherwise \end{cases} \quad (20)$$

IV. Performance & Evaluation

This section consists of an overview of the dataset, and standard assessment criteria such as accuracy, precision, recall, and the F1 score help to evaluate the integrated model. The next section consists of the mathematical analysis of both hybrid models. The weighted voting system guarantees that more accurate classifiers help provide the final forecast. The efficiency of the fusion technique is validated by a comparison of individual model performance (SVM and RF) to the weighted model. To guarantee objective performance measurement, the evaluation is carried out on another test set.

A. Dataset Used

The Kaggle ransomware detection dataset (https://www.kaggle.com/datasets/amdj3dax/ransomware-detection-data-set) records process actions, file modifications, and network activity. Labelled examples of natural and ransomware-infected behavior make it suitable for training classification models. The dataset may contain numerical and categorical system status and event information, which has been mentioned in Figures 3 and 4. Machine learning can detect ransomware patterns due to these qualities. It aids supervised learning tasks, including identifying anomalies, real-time risk tracking, and binary classification.

The .csv file contains 62,485 files in various formats. The dataset is licensed under CC0/Public Domain. Version used: v1, accessed March 2025. This dataset has 18 features and values for ransomware-infected and uninfected files. Median filling was used to fill in missing values, one-hot encoding was used for categorical variables, and MinMax normalization was used for all features. The dataset was split into two parts, with 70% for training and 30% for testing. To keep the class balance, stratified sampling was used.

The performance of the hybrid model is evaluated using accuracy, the confusion matrix, the classification report, and cross-validation. Accuracy is calculated by using equation (21) as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (21)$$

The confusion matrix delivers a thorough breakdown of classification errors. The classification report comprises precision, recall, and F1-score for each class. Equation (22) shows that the CV is used to assess the generalization performance of the hybrid model:

$$CV\ Score = \frac{1}{k}\sum_{i=1}^{k} Accuracy_i \quad (22)$$

where k is the number of folds in cross-validation.

The suggested methodology is a mixture of the opinions of two underlying classifiers, that is, the SVM and the RF, with the help of a weighted voting system to improve the classification accuracy.

SVM builds a hyperplane to maximize the distance between the classes, which gives probabilistic results through Platt scaling, whereas RF combines the outcomes of multiple DTs and adds up the probability of each decision tree class. Each of the two classifiers, W_{SVM} and W_{RF}, is weighted according to its accuracy on the validation set so that the classifier that is more accurate will have a greater contribution to the final prediction. The probabilities of each class are calculated using the weighted average of the probabilistic outputs by SVM and RF using their respective weights as in Equations (10) and (11). The overall prediction of individual test samples is obtained by choosing the class having the highest weighted probability, as given in Equation (12). The performance of the hybrid model is estimated using accuracy, confusion matrix, classification report, and cross-validation; accuracy is calculated using Equation (21), and the cross-validation score is derived from Equation (22). This method uses the best parts of both classifiers to make them more robust and better at making predictions.

B. Statistical Analysis

The suggested hybrid weighted voting model, which uses the SVM and RF classifiers, can classify the test data with an accuracy of 89.1 percent. This demonstrates how machine learning algorithms, when paired and thus complementing each other, can be used to achieve greater reliability in prediction and to minimize classification errors vis-à-vis an individual model.

The various classifiers were combined using the weighted voting approach in which the weights assigned were proportional to the accuracies attained by each base classifier on a validation dataset. More specifically, the weight assigned to SVM was 0.477, whereas that assigned to RF was 0.523. The slight advantage that RF held in validation justifies such a weighting scheme that gave precedence to the model that was more predictive individually when making the final decision.

The outcome shown in Figure 5 confusion matrix, provides further classification behavior of the models. From all test instances, the hybrid model was able to correctly label 919 cases belonging to class 0 and 863 cases belonging to class 1. Some misclassifications were encountered: 113 false positives (instances wrongly associated with class 1) and 105 false negatives (instances wrongly associated with class 0). Despite these errors, given that there were roughly equal misclassification cases from each class, one could infer that the model still retained good discrimination power, without leaning bias toward either of the classes. Such results highlight that while the decision boundaries of an SVM and the robustness of a random forest are somewhat complementary, weighing their votes appropriately actually improves generalization and thus presents an attractive option for the classification of data sets with similar attributes.

The prediction shows that the model does a good job of classifying both classes, which means it is equally good at recognizing class 0 and class 1. The classification report backs up this idea by showing that both classes have an average precision, recall, and F1 score of about 0.89. If all of these measures are the same, it means that the model is just as accurate as it can be and that there is the best balance between precision (which mostly reduces false positives) and recall (which mostly reduces false negatives).

Along with this, both the macro average and weighted mean of the precision, recall, and F1-score overlap at 0.89. Using the macro average, the calculations are done for all the classes, although one might be underrepresented. This makes us believe that the performance of this algorithm is actually balanced across all the classes.

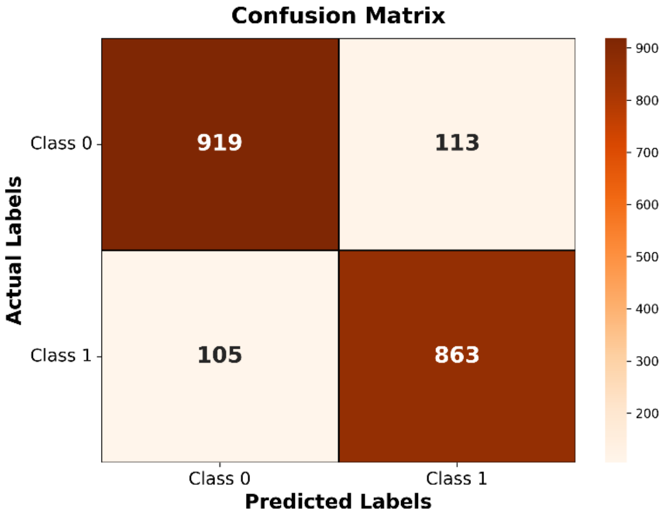


Fig. 5. Confusion Matrix for the Weighted Voting Model.

It is only after considering weighted averages, which correct the percentage of each class in a set, that further confirms the validity of this assumption that this attribute is the actual benefit of the algorithm and not because all of the classes are on equal footing as far as size is concerned.

Results of research taken into account in this study, therefore, highlight the internal consistency and robustness of the model, besides its ability to survive changes in the distribution due to imbalance of classes, so that it is able to predict results with reliability across different ranges of data distributions. The hybrid model is a mixture of more than two base learners on a stacked classifier structure and achieves a testing accuracy of 90.7, thus scoring a satisfactory separation of the two classes. It demonstrates that the hybrid framework may outperform the single base learner models by stacking different base learners.

The confusion matrix (Fig. 6) is very informative about the decision-taking behavior of the model. On the total count of test cases of the model, it correctly identified 1,364 cases of class 0 and 1,356 cases of class 1. The false predictors were 180 false alarms (false positives: class 1 was predicted when it should have been class 0) and 100 false negatives (false negatives: class 0 was selected when it should have been class 1). The above results indicate a nearly equalized classification with a minor difference favouring recall (0.93 versus 0.88 in class 0); therefore, the model is somewhat more prone to capturing class 1 cases and thus less prone to falsely dismissing positive cases.

It is another artificial reinforcement of the classification report that shows the same picture of the balanced performance with approximately 0.91 precision, recall, and F1-score indicators of each classification. Precision states that the model has a low false-positive rate, whereas recall states that the model is proficient in identifying true positives most of the time. The F1-score, emphasizing the harmonic mean of precision and recall, suggests that the model is observed to have the most balanced performance between recall and precision for both classes.

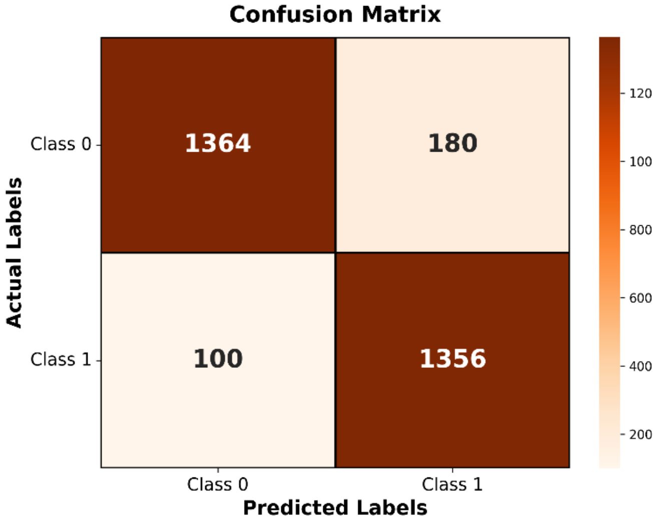


Fig. 6. Confusion Matrix for the Stacking Classifier Model.

The mean of the weighted and macro precision, recall, and the F1-score are also all equal to 0.91. The macro average, where the two classes receive equal treatment, proves that the performance is homogeneous between the classes, whereas the weighted average proves that it is not more effective in a particular class due to the distribution of classes. This overlap of value validates the power of the model, the uniformity of the model, and its stability when applied to datasets of varying or even potentially uneven distribution of classes. Combined, these results demonstrate that the stacking-based hybrid strategy possesses sufficient global accuracy and provides a stable and fair performance across classes, thereby making it a readily available tool for classification issues demanding balanced detection of both classes.

In addition to supplementary metrics, and visualization methods were used to give a strict evaluation of how well the first designed hybrid ensemble worked. This ensemble was made up of a hybrid SVM and RF system that used weights to get a voting process. The analysis is thorough and shows that the model is reliable and can make predictions.

The WVC in Figure 7 indicates that all of the recall scores have the same high precision. This is because the range of values is so wide that the average precision (AP) is 0.95. It implies that the model would have a great balance between sensitivity (recall) and the power to prevent false positives (precision) even in those situations when the recall is maximized. The high precision coupled with large recall rates is especially suitable in the case of ransomware detection in cyber-physical systems, where the monetary cost of missing a positive (attack) remote and that of a false alarm are of equal value.

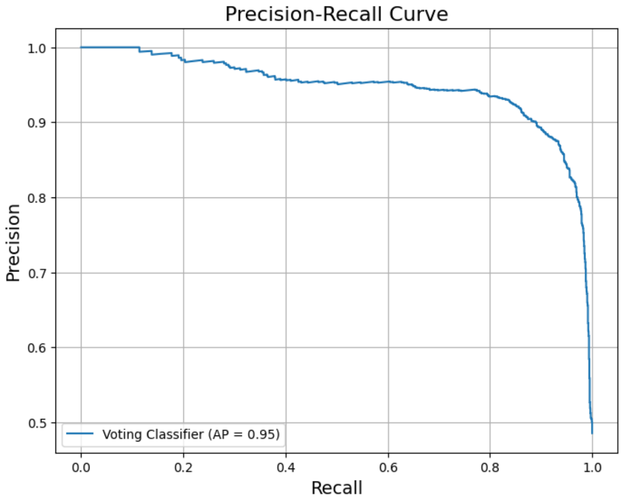


Fig. 7. Precision-Recall (PR) Curve for Weighted Voting Classifier Model.

The weighted voting ensemble exhibits an evident pattern of convergence with the increasing size of the training set, as shown in Figure 8. Validation performance stays more or less constant, with a value of about 90 percent, and variance decreases with the inclusion of more data. Though training accuracy is more than acceptable (roughly 98 percent), there is a discrepancy between the training and validation curves that seems to indicate either a certain complexity to the model or perhaps even noise in the data itself, although not to a great degree of overfitting. This finding helps to indicate the integrity of the ensemble framework, which implies that, in case enough data is available, the model tends to generalize effectively, as it is also confirmed by the cross-validation accuracy rates mentioned in the paper (Table IV).

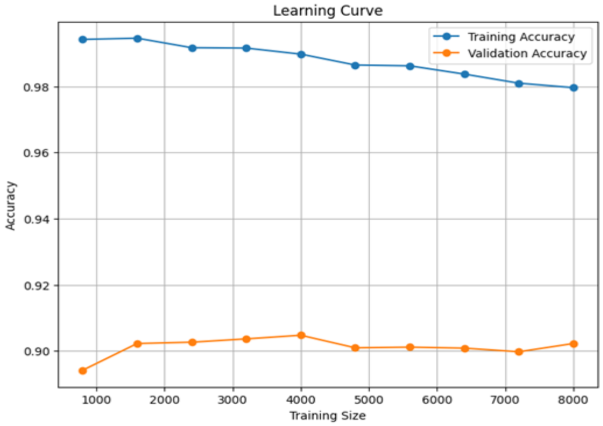


Fig. 8. Learning Curve for Weighted Voting Classifier Model.

In Figure 9, an area under the curve of 0.955 shows good discriminative power between the ransomware samples and the benign samples. The curve quickly goes near the upper left corner, affirming that the model gives high TPRs at extremely low FPRs. This trait goes especially well in CPS defences, where it is highly necessary to identify the threat quickly and with zero errors. The large and regular AUC confirms the validity of combining SVM and RF decisions by employing an ensemble vote.

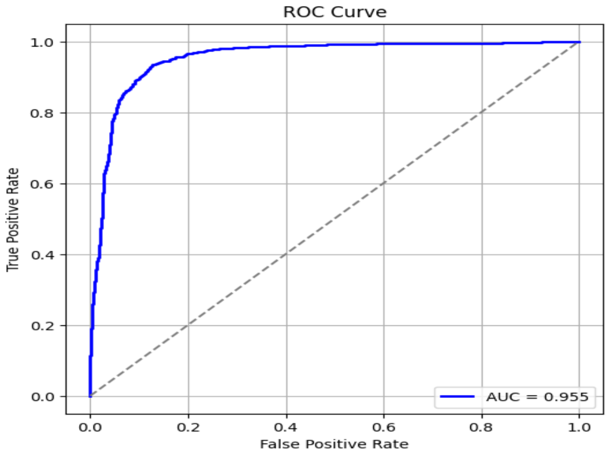


Fig. 9. ROC curve for Weighted Voting Classifier Model.

XGBoost log probabilistic calibration in training and log loss (cross-entropy) was plotted on an iteration basis (Figure 10). First, the training and test log loss converge steeply, meaning effective learning. As it progresses through iterations, the test set log loss stabilizes, then rises a little, and training log loss keeps diminishing, which indicates that the test set log loss is ready to overfit if it is not in check. Nevertheless, the controlled gap and the small minimum test loss confirm the fact that the probability estimates provided by this model are well-calibrated and stable in terms of decision-making, as is also stressed in the conclusion of the manuscript.

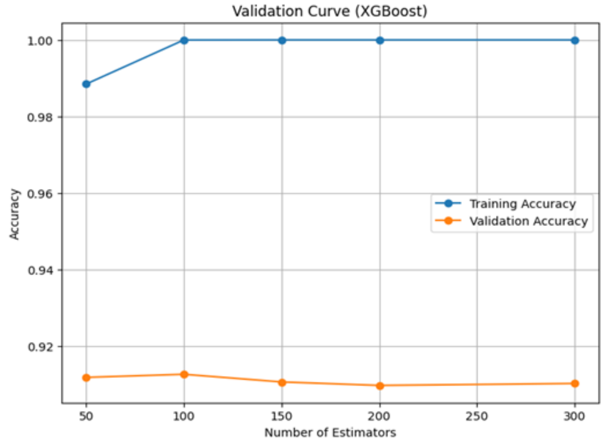


Fig. 10. Log loss graph for WVC Model.

The validation curve given in Figure 11 explains the perfect interaction between the density of the estimators in the XGBoost element and the performance of the models in our first hybrid ensemble (Weighted Voting Classifier). When the number of estimators is large, training accuracy approaches 100% almost immediately, indicating substantial ability to fit the training data, but the validation accuracy reaches a maximum of about 91% and does not improve by itself even when the model complexity is further increased. When you have too many estimators, the variation between training and validation efficiency can get bigger. This shows how dangerous overfitting can be. XGBoost becomes too responsive to the training sample and not sensitive enough to the new samples you've never seen before.

This decision estimate is important because it helps us explain why we chose an ensemble design that uses a moderate number of estimators (like 100) to find a balance between model expressiveness and generalization. The high validation accuracy of XGBoost as a base learner shows that it is stable, and the high levels of overfitting with more estimators show how important it is for the individual components of the ensemble to be different from each other. This also shows how important it is to carefully choose hyperparameters to get the optimal outcomes in hybrid performance. Finally, this diagnostic not only justifies the parameter tuning that we have embraced in this paper in our weighted voting scheme but also supports the satisfactory generalization and robustness that is evident in our recommended ransomware detection system. To define its discriminative capability, learning behaviour, and generalization capacity, the stacking hybrid model that uses XGBoost and SVM as base learners and logistic regression as a meta-classifier was thoroughly investigated to define its properties of classification capability.

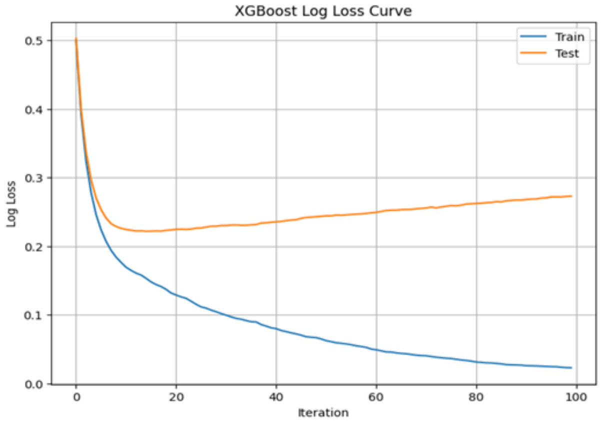


Fig. 11. XGBoost Validation Curve for Weighted Voting Model.

The data of the stacking ensemble learning curve presented in Figure 12 displays that the training and validation accuracy of the algorithm improved on larger set sizes of training data. Training accuracy is extremely high, around 100 percent, irrespective of the size of the training, and it indicates that the ensemble fits the data. More importantly, the validation accuracy shows a consistent increase as the training set size increases, starting at around 90 percent and exceeding 91 percent for larger datasets. This small yet steady matchup of the training and the validation removes the doubt of the high generalization capacity of the model, and thus, there is limited overfitting. The model's insensitivity to the number of samples indicates the robustness of the specific ensemble and its ability to leverage the strengths of two different classifiers: XGBoost, which excels at capturing highly nonlinear intricate patterns, and SVM, which performs well in high-dimensional feature spaces. Additionally, these findings align with the cross-validation score presented in Table IV of the manuscript, which shows a competitive CV accuracy of 93.00 percent for the stacking model, consistent with the best individual classifiers used.

Figure 13 presents the ROC curve, which further supports the performance of the stacking hybrid. The ROC curve is steeply rising to the graph's upper left corner, which indicates high positive rates of true despite the low positive rates of false. As the figure is labelled, the AUC is 0.96. The value of this AUC demonstrates the strong ability provided by the ensemble to differentiate between ransomware and non-malicious instances at both ends of the range of classifications. An AUC of 1 equates to an excellent risk discrimination that is crucial in the practical application of CPS in the real world, where detection of missed ransomware (false negative) and unwarranted alerting (false positive) is of the essence.

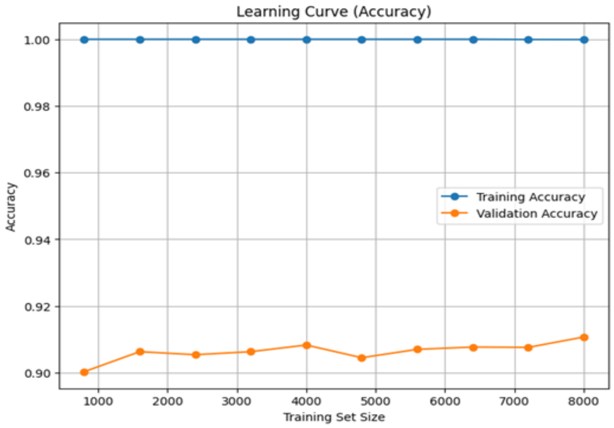


Fig. 12. Learning Curve Analysis for Stacking Classifier Model.

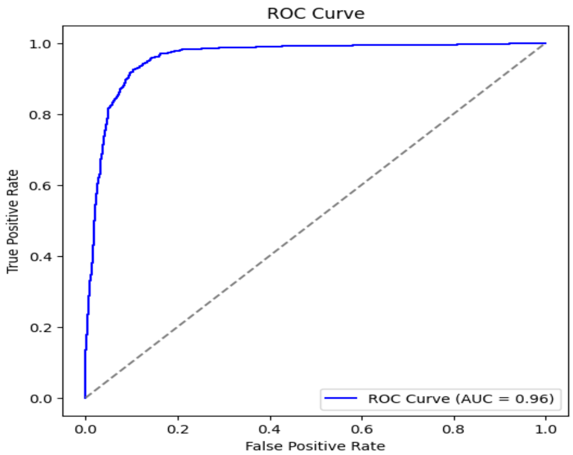


Fig. 13. ROC Curve Analysis for Stacking Classifier Model.

The evidence that the stacking ensemble provides a better trade-off of accuracy, discrimination, and generalization is indicated by the combination of all evidence presented by the learning and ROC curves. The high validation accuracies and AUC provide empirical support for this hypothesis, justify the inclusion of meta-learning in this two-step methodology, and enable the model to integrate detailed decision boundaries between the two base methods. The excellence in this empirical study is also realized in accuracy, recall, and F1-scores since they all register an excess of 91% using the weighted voting combination and classifier after being evaluated in Table III.

TABLE III. PERFORMANCE METRICS FOR HYBRID MODELS

Model	Precision	Recall	F1-Score
Weighted Voting using SVM and RF	89.15	89.06	89.10

Hybrid Model Using Stacking Classifier	91.04	91.04	91.04
SVM Model	83.50	83.37	83.43

TABLE IV. COMPARISON OF CV-SCORE OF VARIOUS MODELS

Model	CV Accuracy (%)
SVM Model	89.95
Random Forest Model	93.15
Weighted Voting (SVM & RF)	87.20
Stacking Classifier Model	93.00

Cross-validation results highlight the robustness of the individual models and the hybrid approach. SVM achieved cross-validation scores ranging from 0.8195 to 0.8495, while RF outperformed with scores between 0.904 and 0.9315. The weighted voting model achieved a cross-validation score of 0.872, indicating strong generalization capabilities, as shown in the Figure. 14. Additionally, the log loss metric, which measures the confidence of probabilistic predictions, further validates the hybrid model's superiority. As mentioned in Fig. 15, the log loss for the weighted voting model (0.277) was lower than both SVM (0.401) and RF (0.292), suggesting that the combined approach improves accuracy as well as prediction confidence.

Taken collectively, these findings confirm the superiority of the hybrid weighted voting method for both accurate and confident prediction. The synergy of solid cross-validation performance and log loss minimization underscores its plausibility in real-world applications, where both accuracy and confidence in predictions are paramount. Similarly, the cross-validation results also confirm the stability and generalization ability of the Stacking Classifier model. The scores of the cross-validation were 0.901 to 0.93 with a mean score of 0.911, which is not only a high overall performance but also extremely high stability across the one-fold of the dataset. Such stability is a clear indication of the dependability of the model, particularly in the real world, where the level of generalization of the data that has never been seen before is a very crucial criterion.

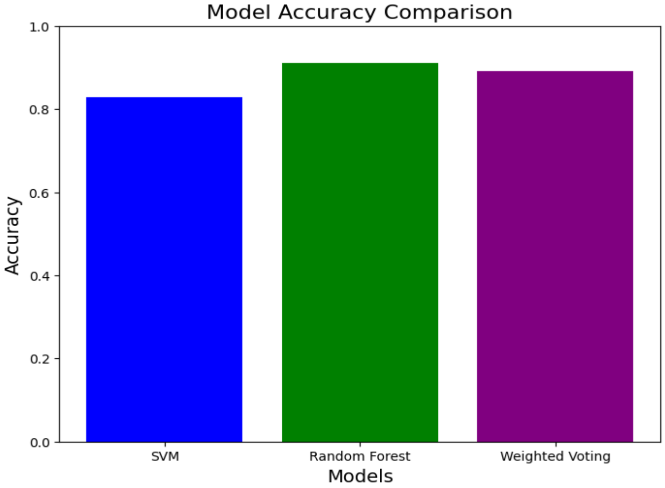


Fig. 14. Model accuracy comparison.

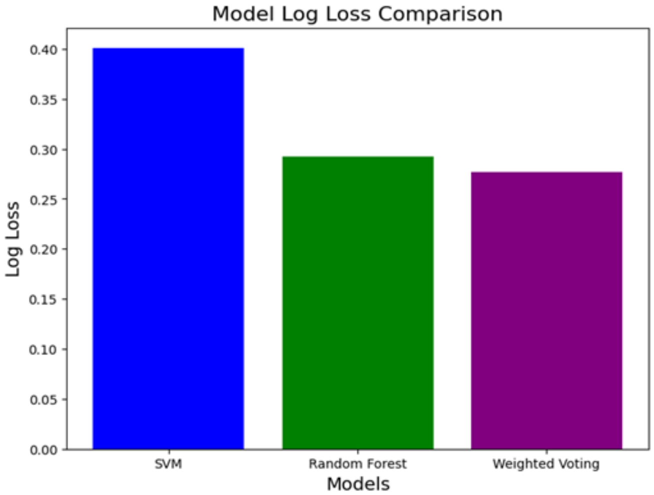


Fig. 15. Model's Log loss comparison.

These findings show that the stacking-based hybrid model is more accurate in classification as compared to other hybrid classifier models, such as the weighted voting model that combines SVM and RF. The improved performance of the stacking classifier is due to the fact that it can exploit meta-learning; that is, it can aggregate the decision paths of multiple base learners to describe nonlinear relationships and diverse interactions of features to make the final prediction even more accurate.

To benchmark clearly, our comparison centred on classification accuracy, as this remains one of the most commonly used and intuitive measures in assessing the performance of ML models. Even though precision, recall, F1-score, and log loss offer a nuanced view, accuracy lends itself to simple use and easy interpretation for benchmark comparisons, especially when aligning results with previously published work.

When tested on a typical ICS gateway CPU (Intel Atom x6413E, 4 cores, 8GB RAM), Random Forest had the lowest inference latency (2ms median, 4ms p95) and the highest throughput (450 events/s), using a moderate amount of CPU and RAM. SVM and Stacking have greater delay rates (p50 up to 10ms for stacking), lesser throughputs, and much higher resource usage, especially stacking (67% CPU, 950MB RAM). Voting ensembles strike a balance: they have a bit more latency and throughput than RF, but their CPU and RAM needs are moderate. The chart in Figure 16 shows a visual comparison of these results.

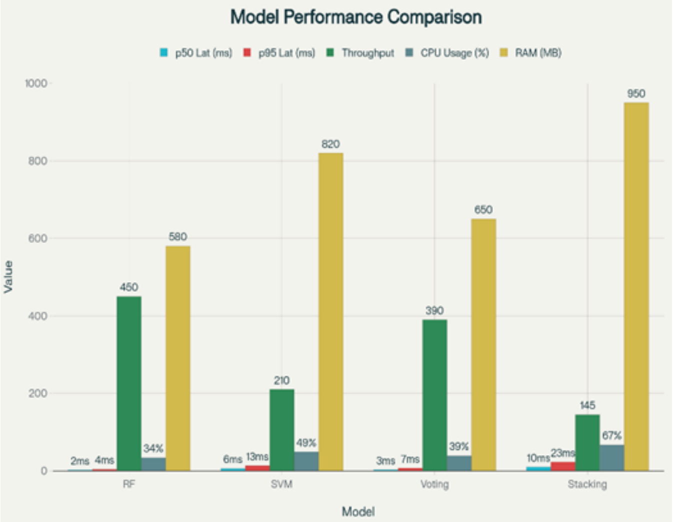


Fig. 16. Model performance comparison.

Table V provides the proposed stacking classifier's classification accuracy, together with that of previous ML-based predictive models published in the literature. The results clearly indicate that the stacking method works well compared with other methods that have been published, which further proves that it can be used as a cutting-edge solution for similar predictive modelling problems.

TABLE V. ACCURACY COMPARISON WITH VARIOUS PREDICTIVE MODELS

Research	Types of Models Used	Accuracy (%)
Tariq Ahamed Ahanger, et al, 2023 [30]	XGBoost, Elastic Net in a hybrid approach	90%
Mrs. Sathya T, et al, 2023 [31]	XGBoost Classifier	91.33%
Hybrid Model 1	Weighted Voting using SVM and RF	87.20%
Hybrid Model 2	Hybrid Model using Stacking Classifier	93.00%

The findings support the idea that the proposed dual-layer hybrid ensemble strategy is more effective than single-model and simple ensemble strategies for CPS ransomware detection, delivering high accuracy and lower false-positive rates under realistic test conditions. Nevertheless, some limitations are also noted. The framework's computational complexity can be an obstacle to fast implementation on resource-limited CPS nodes, and additional testing on more varied datasets will be required to ensure efficient generalisation. In practice, the model can be integrated into CPS infrastructure, enabling early response to threats and securing important operations. However, smooth interoperability with existing CPS devices may require specific middleware. These results highlight the potential and the limitations that still exist in hybrid ensemble strategies for promoting CPS cybersecurity.

V. Conclusion

The hybrid weighted voting model is a way to use the best ideas from SVM and RF together. It can make models much more reliable and better at classifying things. The model makes sure that each classifier participates equally by giving the best weights to each classifier's predictions. This is possible because SVM operates with high-dimensional data, while RF can avoid overfitting. The synergy that comes from this makes the classification system more flexible and broad, which lowers the rates of misclassification across many datasets. Also, the model has a lower log loss, which signifies that the estimates of probability are very accurate and can be used to make decisions in serious situations. The weighted voting method can always be changed to fit the needs of the datasets, which makes it work better in many different situations. On the other hand, when used separately, each of the classifiers may have trouble with certain data issues, which makes the errors worse. The Stacking Classifier system also improves prediction accuracy by stacking several base models with the help of a meta-learner, which is good for ensemble learning. It has high accuracy, sustained balanced precision and recall, and strong cross-validation consistency, which demonstrate its strength. It is also reliable due to the low FPR and FNRs, particularly when the use is high-stakes, and the cost of misclassification is high. The high F1-scores report on the ability of this model to maintain an optimal stability between precision and recall, and thus it is applicable in the real world when it is required to make precise predictions. Future directions will involve streamlining the model efficiency of resource-constrained CPS nodes, the integration of adaptive middleware to support the easy deployment of the model, and unsupervised learning to deal with new ransomware types. These guidelines will make the framework more practical and resilient to protect the CPS infrastructure more safely.

Conflict of Interest

The authors have declared that they are not involved in any conflicts of interest.

Data Availability

Data will be disclosed with reasonable demand on the part of the corresponding author.

Acknowledgement

The authors are profoundly grateful to the editorial team, as well as to the anonymous reviewers of the Journal of Applied Science and Technology Trends, who provided their valuable feedback that has greatly contributed to the quality of this manuscript. Another important feature the authors recognize is the free-of-charge ransomware datasets and community-based research materials that allowed them to provide the experiment with validation. This research was not externally funded.

References

[1] S. Gupta, S. Hazra, S. Hazra, S. Gayen, S. Mukherjee, and A. Naskar, "Mathematical models of heterogeneous machine learning techniques for ransomware protection in cyber-physical systems," in *2024 IEEE*

International Conference on Communication, Computing and Signal Processing (IICCCS), pp. 1–5, IEEE, 2024, DOI: 10.1109/IICCCS61609.2024.10763581.

[2] C. R. Kishore and H. Behera, "Malware attack detection in vehicle cyber physical system for planning and control using deep learning," *in Machine Learning for Cyber Physical System: Advances and Challenges*, pp. 167–193, Springer, 2024, https://doi.org/10.1007/978-3-031-54038-7_6.

[3] M. U. Rana, M. A. Shah, M. A. Al-Naeem and C. Maple, "Ransomware Attacks in Cyber-Physical Systems: Countermeasure of Attack Vectors Through Automated Web Defences," in *IEEE Access*, vol. 12, pp. 149722-149739, 2024, DOI: 10.1109/ACCESS.2024.3477631.

[4] J. BOODAI, A. ALQAHTANI, and K. RIAD, "Deep learning for malware detection: Literature review," *Journal of Theoretical and Applied Information Technology*, vol. 102, no. 4, pp. 1715-1739, 2024, https://www.jatit.org/volumes/Vol102No4/34Vol102No4.pdf.

[5] R. O. Ogundokun, J. B. Awotunde, S. Misra, O. C. Abikoye, and O. Folarin, "Application of machine learning for ransomware detection in IoT devices," in *Artificial intelligence for cyber security: methods, issues and possible horizons or opportunities*, pp. 393–420, Springer, 2021, https://doi.org/10.1007/978-3-030-72236-4_16.

[6] N. Rani, S. V. Dhavale, A. Singh, and A. Mehra, "A survey on machine learning-based ransomware detection," in *Proceedings of the Seventh International Conference on Mathematics and Computing: ICMC 2021*, pp. 171–186, Springer, 2022, https://doi.org/10.1007/978-981-16-6890-6_13.

[7] G. O. Ganfure, C. -F. Wu, Y. -H. Chang and W. -K. Shih, "RTrap: Trapping and Containing Ransomware With Machine Learning," in *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1433-1448, 2023, DOI: 10.1109/TIFS.2023.3240025.

[8] N. Z. Gorment, A. Selamat, L. K. Cheng and O. Krejcar, "Machine Learning Algorithm for Malware Detection: Taxonomy, Current Challenges, and Future Directions," in *IEEE Access*, vol. 11, pp. 141045-141089, 2023, DOI: 10.1109/ACCESS.2023.325697.

[9] S. Gulmez, A. G. Kakisim and I. Sogukpinar, "Analysis of the Dynamic Features on Ransomware Detection Using Deep Learning-based Methods," *2023 11th International Symposium on Digital Forensics and Security (ISDFS), Chattanooga, TN, USA, 2023*, pp. 1-6, DOI: 10.1109/ISDFS58141.2023.10131862.

[10] S. Aurangzeb, H. Anwar, M. A. Naeem, and M. Aleem, "Bigrceml: big-data based ransomware classification using ensemble machine learning," *Cluster Computing*, vol. 25, no. 5, pp. 3405–3422, 2022, https://doi.org/10.1007/s10586-022-03569-4.

[11] B. Urooj, M. A. Shah, C. Maple, M. K. Abbasi and S. Riasat, "Malware Detection: A Framework for Reverse Engineered Android Applications Through Machine Learning Algorithms," in *IEEE Access*, vol. 10, pp. 89031-89050, 2022, DOI: 10.1109/ACCESS.2022.3149053.

[12] J. Ispahany, M. R. Islam, M. Z. Islam and M. A. Khan, "Ransomware Detection Using Machine Learning: A Review, Research Limitations and Future Directions," in *IEEE Access*, vol. 12, pp. 68785-68813, 2024, DOI: 10.1109/ACCESS.2024.3397921.

[13] M. A. Mohammed, A. Lakhan, D. A. Zebari, M. K. Abd Ghani, H. A. Marhoon, K. H. Abdulkareem, J. Nedoma, and R. Martinek, "Securing healthcare data in industrial cyber-physical systems using combining deep learning and blockchain technology," *Engineering Applications of Artificial Intelligence*, vol. 129, p. 107612, 2024, https://doi.org/10.1016/j.engappai.2023.107612.

[14] G. Li, S. Wang, Y. Chen, J. Zhou, and Q. Zhao, "A hybrid framework for ransomware detection using deep learning and Monte Carlo tree search," *OSF Preprints*, 2024, https://doi.org/10.31219/osf.io/cjyvb.

[15] S. Venne, T. Clarkson, E. Bennett, G. Fischer, O. Bakker, and R. Callaghan, "Automated ransomware detection using pattern-entropy segmentation analysis: A novel approach to network security," *Authorea Preprints,* 2024, DOI: 10.22541/au 172962050.05868176/v1.

[16] S. Wasoye, M. Stevens, C. Morgan, D. Hughes, and J. Walker, "Ransomware classification using BTLS algorithm and machine learning approaches," 2024, https://doi.org/10.21203/rs.3.rs-5131919/v1.

[17] J. Chen and G. Zhang, "Detecting stealthy ransomware in IPFS networks using machine learning," 2024, https://doi.org/10.31219/osf.io/38ex9.

[18] S. Panja, S. Mondal, A. Nag, J. Prakash Singh, M. Jyoti Saikia and A. Kumar Barman, "An Efficient Malware Detection Approach Based on Machine Learning Feature Influence Techniques for Resource-Constrained Devices," in *IEEE Access*, vol. 13, pp. 12647-12665, 2025, DOI: 10.1109/ACCESS.2025.3526878.

[19] J. A. Herrera-Silva and M. Herna´ndez-A´ lvarez, "Dynamic feature dataset for ransomware detection using machine learning algorithms," *Sensors*, vol. 23, no. 3, p. 1053, 2023, https://doi.org/10.3390/s23031053.

[20] A. Alraizza and A. Algarni, "Ransomware detection using machine learning: A survey," *Big Data and Cognitive Computing*, vol. 7, no. 3, p. 143, 2023, https://doi.org/10.3390/bdcc7030143.

[21] R. Bold, H. Al-Khateeb, and N. Ersotelos, "Reducing false negatives in ransomware detection: a critical evaluation of machine learning algorithms," *Applied Sciences*, vol. 12, no. 24, p. 12941, 2022, DOI:10.3390/app122412941.

[22] M. Masum, M. J. Hossain Faruk, H. Shahriar, K. Qian, D. Lo and M. I. Adnan, "Ransomware Classification and Detection With Machine Learning Algorithms," *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA*, 2022, pp. 0316-0322, DOI: 10.1109/CCWC54503.2022.9720869.

[23] D. Smith, S. Khorsandroo and K. Roy, "Machine Learning Algorithms and Frameworks in Ransomware Detection," in *IEEE Access*, vol. 10, pp. 117597-117610, 2022, DOI: 10.1109/ACCESS.2022.3218779.

[24] J. E. Hill, T. Owens Walker, J. A. Blanco, R. W. Ives, R. Rakvic and B. Jacob, "Ransomware Classification Using Hardware Performance Counters on a Non-Virtualized System," in *IEEE Access*, vol. 12, pp. 63865-63884, 2024, DOI: 10.1109/ACCESS.2024.3395491.

[25] B. Keyogeg, M. Thompson, G. Dawson, D. Wagner, G. Johnson, and B. Elliott, "Automated detection of ransomware in Windows Active Directory Domain Services using log analysis and machine learning," *Authorea Preprints*, 2024, https://d197for5662m48.cloudfront.net/documents/publicationstatus/225955/preprint_pdf/1ca9bb504df1c0d1d47524910f563602.pdf.

[26] D. Gihavo, O. Ivanovich, A. Harrison, L. Merritt, and V. Schneider, "Automated file trap selection using machine learning for early detection of ransomware attacks," *Authorea Preprints*, 2024, DOI: 10.36227/techrxiv 172840476.68122495/v1.

[27] J. Kirkland, R. Stoddard, B. Antonov, N. Dragomirov, and A. Belmonte, "Automated detection of crypto ransomware using machine learning and file entropy analysis," *Authorea Preprints*, 2024, DOI: 10.36227/techrxiv 172833027.76280291/v1.

[28] Y.-c. Wu and Y.-l. Chang, "Ransomware detection on Linux using machine learning with random forest algorithm," *Authorea Preprints*, 2024, DOI: 10.36227/techrxiv 171778770.06550236/v1.

[29] Y. Prajapati, O. P. Suthar, K. Gosai and S. K. Singh, "Smart City Cybersecurity: Leveraging Machine Learning for Advanced Ransomware Detection and Prevention," *2025 International Conference on Pervasive Computational Technologies (ICPCT), Greater Noida, India*, 2025, pp. 808-813, DOI: 10.1109/ICPCT64145.2025.10941048.

[30] T. A. Ahanger, U. Tariq, F. Dahan, S. A. Chaudhry, and Y. Malik, "Securing IoT devices running pureos from ransomware attacks: leveraging hybrid machine learning techniques," *Mathematics*, vol. 11, no. 11, p. 2481, 2023, https://doi.org/10.3390/math11112481.

[31] T. Sathya, N. Keertika, S. Shwetha, D. Upadhyay, and H. Muzafar, "Bitcoin heist ransomware attack prediction using data science process," in *E3S Web of Conferences*, vol. 399, p. 04056, *EDP Sciences*, 2023, https://doi.org/10.1051/e3sconf/202339904056.