Identifying Ransomware Attacks through Processor and Disk Usage Analysis

Mrs. R. JYOSNA DEVI[1], G.YASASWINI[2]

[1] Associate Professor of MCA, Dept of MCA, Audisankara College of Engineering and Technology (AUTONOMOUS) Gudur (M), Tirupati (Dt), AP

[2]PG Scholar, Dept of MCA, Audisankara College of Engineering and Technology (AUTONOMOUS) Gudur (M), Tirupati (Dt), AP

Abstract: The project addresses the challenge of ransomware detection, acknowledging the limitations of current approaches involving process monitoring and data analysis. The aim is to develop a robust and practical detection method for ransomware executed on a virtual machine (VM). Data collection focuses on specific processor and disk I/O events for the entire VM from the host machine. Leveraging machine learning (ML), particularly a random forest (RF) classifier, the project aims to create an effective detection model. This approach minimizes monitoring overhead and mitigates the risk of data contamination by ransomware. The proposed method demonstrates resilience to variations in user workloads, overcoming a common challenge in ransomware detection. By avoiding continuous monitoring of every process on the target machine, the model remains adaptable to different user scenarios. The project's effectiveness is measured across various user workloads and 22 ransomware samples. This Project contributes a practical and efficient solution to the ongoing ransomware threat by providing a reliable detection model. By utilizing selected processor and disk I/O events and incorporating machine learning, the project minimizes monitoring overhead, enhances detection speed, and ensures adaptability to evolving ransomware variants. In this project additional enhancements were introduced, incorporating Convolutional Neural Network 2D (CNN2D) and an ensemble model with a voting classifier to further improve ransomware detection accuracy. The voting classifier, comprising multiple machine learning classifiers, demonstrated a remarkable 99% accuracy in making final predictions, showcasing the effectiveness of combining different models for robust detection.

Index terms - *Deep learning, disk statistics, hardware performance counters, machine learning, ransomware, virtual machines.*

1. INTRODUCTION

Ransomware is malware that encrypts files on a target computer or locks the computer to render the target machine and its data unusable. Cyber attackers use ransomware attacks to extort victims' money. Nation-state actors may use ransomware attacks to inflict harm on the critical infrastructure of their adversaries. These attacks are often combined with the exfiltration of victims' data to compel the victim to pay a ransom or sell the data on the dark web. In 2022 around 70% of businesses were victimized by ransomware attacks [1]. Ransomware is expected to attack a business, consumer, or device every 2 seconds by 2031, up from every 11 seconds in 2021. The damage cost was $20 billion in 2021 and is likely to exceed $265 Billion By 2031 [2]. Several

researchers have recently investigated the detection of ransomware attacks.

Signature-based detection [3], [4] relies on the hash values generated by antivirus software for known ransomware and checks the target machine for files that match the hash values. However, polymorphic and metamorphic versions of previously known ransomware can bypass such signature-based detection [4], [5]. Therefore, behavioral or runtime detection of ransomware in execution complements signature-based detection methods. The behavioral analysis is a dynamic analysis that looks into the virus's behavior—the sequence of actions performed by the ransomware after it infects the victim machine. While malware activities vary widely, ransomware must perform a specific sequence of activities to encrypt as many data files as possible quickly. Some recent ransomware such as LockBit2.0, Darkside, and BlackMatter encrypt only parts (the first few bytes) of files to render more files unusable quickly [6]. Therefore, the requirement of quickly encrypting user files is likely to differentiate ransomware's runtime behavior from that of a benign application. The premise is that a system under ransomware attack must exhibit some form of immutable anomalous behavior. For example, the ransomware must access files from the hard disk and use the processor for data encryption resulting in elevated activity; suitably trained machine-learning methods may detect the elevated activity.

Runtime detection implemented on the target machine requires continual monitoring of various processes or components and subsystems, collecting data related to various events, and analyzing the data for anomalous behavior [7], [8]. Ransomware may try camouflaging its runtime behavior by creating additional processes and activities. However, the fact remains that a system under attack exhibits some form of elevated activity, which is detectable with appropriate analysis. Runtime detection is resource-intensive and intrusive if the monitoring is done on the target machine since the process corresponding to ransomware may not be easy to identify, and multiple processes must be monitored. Also, such monitoring is prone to be disabled by the ransomware designed to shut down active processes before encrypting files.

Hardware performance counters (HPCs) are specialpurpose registers that count processor and system events per process or the entire system. The current processors can count hundreds of processor and system events, such as the number of instructions executed, cache misses, and accesses to off-chip memory. The data collected using HPCs are frequently used for performance analysis and tuning of the system software. However, several recent research efforts have investigated their use for malware detection [9], [10], [11], [12], [13], [14]. Alam et al. [15] utilized HPC data collected for each process running on the system. However, monitoring many processes is impractical as it can cripple the system's performance. Pundir et al. [7] collected the data at the machine level. However, their experiments are limited to a single workload of a Windows virtual machine (VM), and a change in the workload of a VM (varying the number of applications) may significantly impact the detection accuracy.

2. LITERATURE SURVEY

Malware, such as Trojan Horse, Worms and Spy ware severely threatens Internet. We observed that although malware and its variants may vary a lot from content signatures, they share some behavior

features at a higher level which are more precise in revealing the real intent of malware. [4] This paper investigates the technique of malware behavior extraction, presents the formal [3, 5, 9, 10, 12] Malware Behavior Feature (MBF) extraction method, and proposes the malicious behavior feature based malware detection algorithm. Finally we designed and implemented the MBF based malware detection system, and the experimental results show that it can detect newly appeared unknown malwares.

Among many prevailing malware, crypto-ransomware poses a significant threat as it financially extorts affected users by creating denial of access via unauthorized encryption of their documents as well as holding their documents hostage and financially extorting them. This results in millions of dollars of annual losses worldwide. Multiple variants of ransomware are growing in number with capabilities of evasion from many anti-viruses and software-only malware detection schemes that rely on static execution signatures. [7] In this paper, we propose a hardware-assisted scheme, called RanStop, for early detection of crypto-ransomware infection in commodity processors. RanStop leverages the information of hardware performance counters embedded in the performance monitoring unit in modern processors to observe micro-architectural event sets and detects known and unknown crypto-ransomware variants. In this paper, we train a recurrent neural network-based machine learning architecture using long short-term memory (LSTM) [15, 52, 54] model for analyzing micro-architectural events in the hardware domain when executing multiple variants of ransomware as well as benign programs. We create timeseries to develop intrinsic statistical features using the information of related HPCs and improve the detection accuracy of RanStop

and reduce noise by via LSTM [52,54] and global average pooling. As an early detection scheme, RanStop can accurately and quickly identify ransomware within 2ms from the start of the program execution by analyzing HPC information collected for 20 timestamps each 100us apart. This detection time is too early for a ransomware to make any significant damage, if none. Moreover, validation against benign programs with behavioral (sub-routine-centric) similarity with that of a crypto-ransomware shows that RanStop can detect ransomware with an average of 97% accuracy for fifty random trials.

Ransomware has recently (re)emerged as a popular malware that targets a wide range of victims - from individual users to corporate ones for monetary gain. Our key observation on the existing ransomware detection mechanisms is that they fail to provide an early warning in real-time which results in irreversible encryption of a significant number of files while the post-encryption techniques (e.g., key extraction, file restoration) suffer from several limitations. [27], [28] Also, the existing detection mechanisms result in high false positives being unable to determine the original intent of file changes, i.e., they fail to distinguish whether a significant change in a file is due to a ransomware encryption or due to a file operation by the user herself (e.g., benign encryption or compression). To address these challenges, in this paper [8], we introduce a ransomware detection mechanism, RWGuard, which is able to detect crypto-ransomware in real-time on a user's machine by (1) deploying decoy techniques, (2) carefully monitoring both the running processes and the file system for malicious activities, and (3) omitting benign file changes from being flagged through the learning of users'

encryption behavior. We evaluate our system against samples from 14 most prevalent ransomware families to date [22], [23], [24], [25], [26]. Our experiments show that RWGuard is effective in real-time detection of ransomware with zero false negative and negligible false positive ($\sim$ 0.1%) rates while incurring an overhead of only $\sim$ 1.9%.

The proliferation of computers in any domain is followed by the proliferation of malware in that domain. Systems, including the latest mobile platforms, are laden with viruses, rootkits, spyware, adware and other classes of malware. Despite the existence of anti-virus software, malware threats persist and are growing as there exist a myriad of ways to subvert anti-virus (AV) software. In fact, attackers today exploit bugs in the AV software to break into systems. In this paper [9], we examine the feasibility of building a malware detector in hardware using existing performance counters. We find that data from performance counters can be used to identify malware and that our detection techniques are robust to minor variations in malware programs. As a result, after examining a small set of variations within a family of malware on [33, 36] Android ARM and Intel Linux platforms, we can detect many variations within that family. Further, our proposed hardware modifications allow the malware detector to run securely beneath the system software, thus setting the stage for AV implementations that are simpler and less buggy than software AV. Combined, the robustness and security of hardware AV techniques have the potential to advance state-of-the-art online malware detection.

Recent works have shown promise in using microarchitectural execution patterns to detect malware programs. These detectors belong to a class of detectors known as signature-based detectors as they catch malware by comparing a program's execution pattern (signature) to execution patterns of known malware programs. In this work [10], we propose a new class of detectors - anomaly-based hardware malware detectors - that do not require signatures for malware detection [9], [10], [11], [12], [13], [14], and thus can catch a wider range of malware including potentially novel ones. We use unsupervised machine learning to build profiles of normal program execution based on data from performance counters, and use these profiles to detect significant deviations in program behavior that occur as a result of malware exploitation. We show that real-world exploitation of popular programs such as IE and Adobe PDF Reader on a Windows/x86 platform can be detected with nearly perfect certainty. We also examine the limits and challenges in implementing this approach in face of a sophisticated adversary attempting to evade anomaly-based detection. The proposed detector is complementary to previously proposed signature-based detectors and can be used together to improve security.

3. METHODOLOGY

i) Proposed Work:

The proposed system introduces a novel approach to ransomware detection on virtual machines (VMs). It collects specific processor and disk I/O events for the entire VM from the host machine. Machine learning, particularly a random forest (RF) classifier [52], is employed to develop a robust detection model. This method aims to avoid the monitoring overhead associated with continuous monitoring of every process on the target machine, reducing the risk of

data contamination by ransomware. It also demonstrates resilience to variations in user workloads. The proposed system achieves fast detection with high accuracy for both known and unknown ransomware, with the [52] RF classifier outperforming other tested classifiers. In this paper additional enhancements were introduced, incorporating Convolutional Neural Network 2D (CNN2D) and an ensemble model with a voting classifier to further improve ransomware detection accuracy. The voting classifier, comprising multiple machine learning classifiers, demonstrated a remarkable 99% accuracy in making final predictions, showcasing the effectiveness of combining different models for robust detection.

ii) System Architecture:

This paper investigates the fast detection of ransomware in execution on a Windows 10 virtual machine (VM). We collect both the HPC and disk I/O data at the host-machine level. The target (VM) is ignorant of the monitoring and data collection; also, there is little or no impact on its performance. [24] We use machine learning (ML)-based models to analyze the data and detect ransomware in execution [52]. Our method is particularly suited for protecting users of VMs in a cloud environment. We present an approach to detect ransomware accurately using HPC and disk I/O data observed from the host machine. Our approach avoids the overhead of monitoring many processes on the target machine and prevents data contamination by the ransomware designed to thwart such monitoring activities.
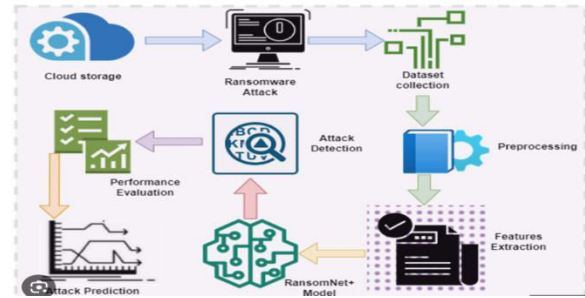
Fig 1 Proposed architecture

iii) Dataset collection:

The HPC dataset utilized in the project consists of records capturing processor and disk I/O events during the execution of virtual machines. This dataset is carefully curated to represent a diverse range of system activities, providing a comprehensive foundation for training and testing ransomware detection models. With both known ransomware samples for model calibration and unknown samples for robustness testing, the HPC dataset facilitates a realistic simulation of potential ransomware behaviors in real-world computing environments [16], [17], [18], [19], [20].

Fig 2 Dataset

iv) Data Processing:

Data processing involves transforming raw data into valuable information for businesses. Generally, data scientists process data, which includes collecting, organizing, cleaning, verifying, analyzing, and converting it into readable formats such as graphs or

documents. Data processing can be done using three methods i.e., manual, mechanical, and electronic. The aim is to increase the value of information and facilitate decision-making. This enables businesses to improve their operations and make timely strategic decisions. Automated data processing solutions, such as computer software programming, play a significant role in this. It can help turn large amounts of data, including big data, into meaningful insights for quality management and decision-making.

v) Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling.

Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important.

vi) Algorithms:

Long Short Term Memory (LSTM): LSTM is a type of recurrent neural network (RNN) designed to overcome the vanishing gradient problem in

traditional RNNs. It introduces a memory cell that allows the model to capture long-term dependencies in sequential data, making it well-suited for tasks involving time series or sequential patterns. [15] LSTMs are likely used in the project for their ability to model and understand temporal dependencies, crucial in ransomware detection where the sequence of system events and behaviors plays a significant role. [45] LSTMs can capture nuanced patterns over time, enhancing the model's ability to detect ransomware activities.

Fig 3 LSTM

Deep Neural Network (DNN): A Deep Neural Network is a type of artificial neural network with multiple hidden layers between the input and output layers. These networks are capable of learning intricate hierarchical representations of data, making them suitable for complex tasks requiring feature abstraction and representation. DNNs might be employed in the project for their capacity to learn intricate features and relationships within the collected data. In ransomware detection, where subtle and complex patterns may exist, DNNs can provide a powerful framework for feature extraction and learning high-level representations [8], [13], [14].

Fig 4 DNN

XGBoost: XGBoost (Extreme Gradient Boosting) is a machine learning algorithm that belongs to the family of gradient boosting techniques. It builds an ensemble of weak learners (typically decision trees) in a sequential manner, with each tree correcting the errors of its predecessor, leading to a robust and accurate model. XGBoost is likely incorporated for its efficiency in handling both classification tasks and large datasets. In the context of ransomware detection, XGBoost can offer strong predictive capabilities, effectively capturing the diverse characteristics of ransomware behaviors and aiding in the creation of an accurate detection model [8], [13], [14].

Fig 5 Xgboost

Random Forest: Random Forest is an ensemble learning algorithm that constructs a multitude of decision trees during training. It outputs the mode of the classes (classification) or the mean prediction (regression) of individual trees. Usage in the Project:

Random Forest is employed for its ability to handle complex classification tasks. In ransomware detection, where diverse patterns may exist, a Random Forest ensemble can enhance accuracy by combining the strengths of multiple decision trees, providing a robust and reliable model.

Fig 6 Random forest

Decision Tree: A Decision Tree is a tree-like model where each node represents a decision based on the input features. It recursively splits the dataset into subsets, leading to terminal nodes that correspond to the final decision or prediction. Decision Trees are utilized for their interpretability and simplicity in representing decision-making processes. In the context of ransomware detection, Decision Trees can provide insights into the sequential steps leading to a decision, aiding in understanding the factors contributing to the detection outcome [52].

Fig 7 Decision tree

K – Nearest Neighbor (KNN):

KNN is a supervised machine learning algorithm used for classification and regression tasks. It classifies a new data point based on majority class vote or predicts its value by averaging the values of its k nearest neighbors in the feature space. KNN is likely used for its simplicity and effectiveness in capturing local patterns in the data. In ransomware detection, where subtle variations may exist, KNN can provide a flexible approach for identifying similar patterns in the dataset.

Fig 8 KNN

Support Vector Machine (SVM):

SVM is a supervised machine learning algorithm used for classification and regression tasks. It finds a hyperplane that best separates data into different classes or predicts a continuous outcome, maximizing the margin between classes. SVM is employed for its capability to handle high-dimensional data and find optimal decision boundaries. In ransomware detection, where feature spaces can be complex, SVM can provide a robust method for effective classification by identifying clear decision boundaries [52].

Fig 9 SVM

2D Convolutional Neural Network (CNN2D):

CNN2D is a deep learning algorithm designed for processing grid-like data, typically used for image analysis. However, in this project, CNN2D is adapted to handle continuous data, utilizing convolutional layers to automatically learn hierarchical features and patterns. CNN2D is employed for its ability to automatically extract complex features from continuous data. In the context of ransomware detection, where patterns in the sequential and continuous system activities are crucial, CNN2D can capture intricate features, contributing to a more effective detection model.

Fig 10 CNN2D

Voting Classifier:

A Voting Classifier combines predictions from multiple individual models using majority voting or averaging. It is employed to enhance overall model performance by leveraging diverse algorithms. The Voting Classifier integrates predictions from various algorithms,. This ensemble approach improves the overall ransomware detection system's robustness and accuracy by considering diverse perspectives from different models.

```
Voting Classifier
from sklearn.ensemble import RandomForestClassifier, VotingClassifier, AdaBoostClassifier
clf1 = AdaBoostClassifier(n_estimators=10, random_state=0)
clf2 = RandomForestClassifier(n_estimators=5, random_state=1)

eclf = VotingClassifier(estimators=[('ad', clf1), ('rf', clf2)], voting='soft')
eclf.fit(X_train, y_train)

predict = eclf.predict(X_test)
calculateMetrics("Voting Classifier", predict, y_test)

rf_acc = accuracy_score(predict, y_test)
rf_prec = precision_score(predict, y_test,average='macro')
rf_rec = recall_score(predict, y_test,average='macro')
rf_f1 = f1_score(predict, y_test,average='macro')

storeResults('Voting Classifier',rf_acc,rf_prec,rf_rec,rf_f1)
```

Fig 11 Voting classifier

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$
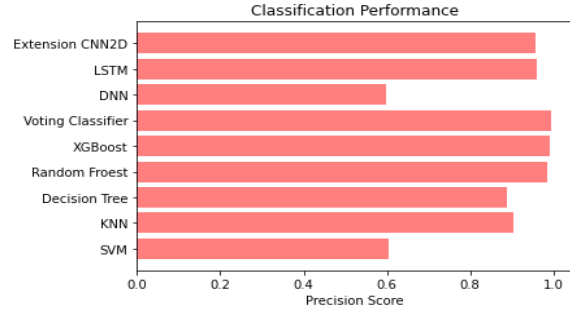
Fig 12 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$Recall = \frac{TP}{TP + FN}$$

Fig 13 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

Fig 14 Accuracy graph

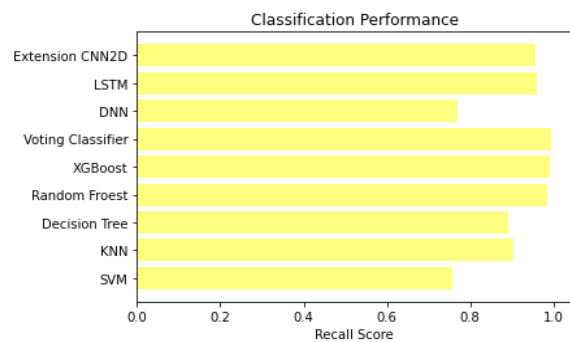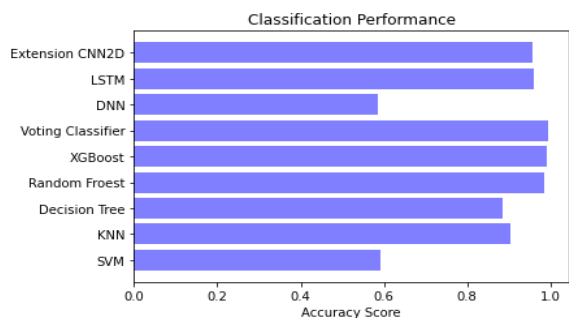**F1 Score:** The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$F1\ Score\ = 2 * \frac{Recall\ \times Precision}{Recall + Precision} * 100$$
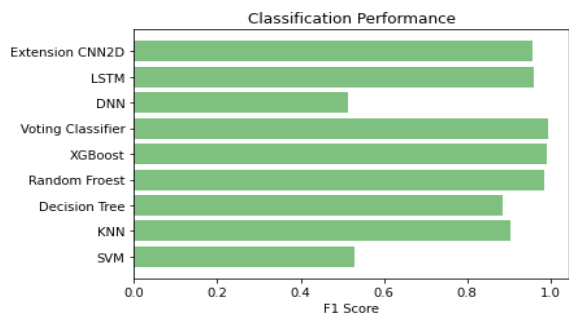
Fig 15 F1Score

ML Model	Accuracy	Precision	Recall	F1-score
SVM	0.593	0.605	0.758	0.529
KNN	0.904	0.905	0.905	0.904
Decision Tree	0.885	0.887	0.891	0.885
Random Forest	0.985	0.985	0.985	0.985
XGBoost	0.992	0.991	0.992	0.992
Extension Voting Classifier	0.995	0.995	0.995	0.995
DNN	0.585	0.597	0.769	0.513
LSTM	0.960	0.961	0.961	0.960
Extension CNN2D	0.956	0.957	0.957	0.956

Fig 16 Performance Evaluation

Fig 17 Home page

Fig 18 Signin page

Fig 19 Login page

Fig 20 User input

Fig 21 Predict result for given input

5.   CONCLUSION

The project successfully introduces an innovative approach to ransomware detection [9], [10], [11], [12], [13], [14]., utilizing virtualization technology, hardware performance counters, and IO events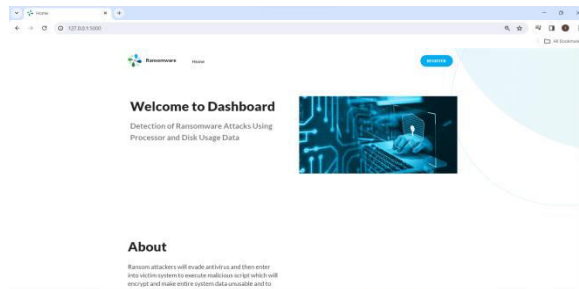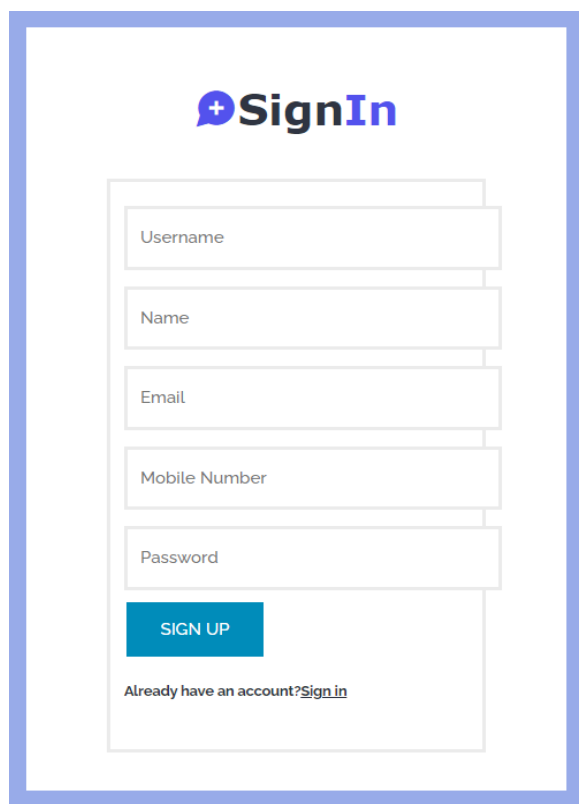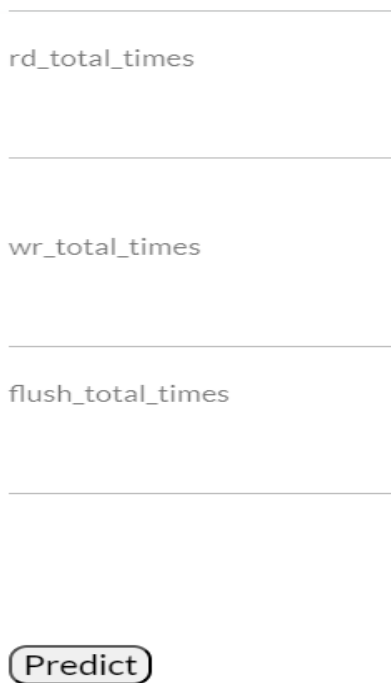 data to enhance accuracy while minimizing system performance impact. Through extensive experimentation, the project evaluates various machine learning algorithms, including SVM, KNN, Decision Tree, Random Forest, XGBOOST, DNN, and LSTM, revealing that Random Forest and XGBOOST [52] consistently demonstrate high accuracy in predicting ransomware activities. The project explores the efficacy of deep learning models, specifically DNN and LSTM, contributing valuable insights into their performance in comparison to traditional machine learning algorithms, further enriching the diversity of predictive approaches. The project contributes to the cybersecurity community by publishing a publicly accessible dataset derived from various programs, fostering collaboration and enabling researchers to benchmark their ransomware detection models. The project seamlessly integrates Flask for web framework and SQLite for user registration and authentication, providing a user-friendly interface where users can input data, have it preprocessed, and obtain predictions from the trained model, enhancing practical applicability.

6.   FUTURE SCOPE

Further investigation could delve into assessing the efficacy of the proposed method in identifying novel and emerging ransomware variants, given that the present study concentrated on a mix of both known and unknown ransomware. Extending the project to evaluate how diverse user workloads impact

ransomware detection [7] would provide valuable insights, building on the study's demonstrated adaptability to varying workloads. Voting classifier which is extension to the project , has demonstrated exceptional performance with a 99% accuracy in ransomware detection. Rigorous testing on the front end using feature values substantiates its robustness and effectiveness in reliably identifying and mitigating ransomware threats. Implementing and testing the proposed approach in real-world scenarios would offer practical insights into its effectiveness in identifying ransomware attacks within live production environments. The project has the potential to enhance its ransomware detection [24, 25, 34] capabilities by investigating the integration of additional data sources or features into the machine learning model. Collaboration with cybersecurity experts and organizations presents an opportunity to validate findings and refine the proposed approach based on real-world expertise and insights.

REFERENCES

[1] SR Department. (2022). Ransomware victimization rate 2022. Accessed: Apr. 6, 2022. [Online]. Available: https://www.statista.com/statistics/204457/businesses-ransomware-attack-rate/

[2] D. Braue. (2022). Ransomware Damage Costs. Accessed: Sep. 16, 2022. [Online]. Available: https://cybersecurityventures.com/globalransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/

[3] Logix Consulting. (2020). What is Signature Based Malware Detection. Accessed: Apr. 3, 2023. [Online]. Available: https://www.logixconsulting.com/2020/12/15/what-is-signature-based-malware-detection/

[4] W. Liu, P. Ren, K. Liu, and H.-X. Duan, ''Behavior-based malware analysis and detection,'' in Proc. 1st Int. Workshop Complex. Data Mining, Sep. 2011, pp. 39–42.

[5] (2021). Polymorphic Malware. Accessed: Apr. 3, 2023. [Online]. Available: https://www.thesslstore.com/blog/polymorphic-malware-andmetamorphic-malware-what-you-need-to-know/

[6] M. Loman. (2021). Lockfile Ransomware's Box of Tricks: Intermittent Encryption and Evasion. Accessed: Nov. 16, 2021. [Online]. Available: https://news.sophos.com/en-us/2021/08/27/lockfile-ransomwares-box-oftricks-intermittent-encryption-and-evasion/

[7] N. Pundir, M. Tehranipoor, and F. Rahman, ''RanStop: A hardwareassisted runtime crypto-ransomware detection technique,'' 2020, arXiv:2011.12248.

[8] S. Mehnaz, A. Mudgerikar, and E. Bertino, ''RWGuard: A real-time detection system against cryptographic ransomware,'' in Proc. Int. Symp. Res. Attacks, Intrusions, Defenses. Cham, Switzerland: Springer, 2018, pp. 114–136.

[9] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Sethumadhavan, and S. Stolfo, ''On the feasibility of online malware detection with performance counters,'' ACM SIGARCH Comput. Archit. News, vol. 41, no. 3, pp. 559–570, Jun. 2013.

[10] A. Tang, S. Sethumadhavan, and S. J. Stolfo, ''Unsupervised anomalybased malware detection

using hardware features,'' in Proc. Int. Workshop Recent Adv. Intrusion Detection. Cham, Switzerland: Springer, 2014, pp. 109–129.

[11] S. Das, J. Werner, M. Antonakakis, M. Polychronakis, and F. Monrose, ''SoK: The challenges, pitfalls, and perils of using hardware performance counters for security,'' in Proc. IEEE Symp. Secur. Privacy (SP), May 2019, pp. 20–38.

[12] S. P. Kadiyala, P. Jadhav, S.-K. Lam, and T. Srikanthan, ''Hardware performance counter-based fine-grained malware detection,'' ACM Trans. Embedded Comput. Syst., vol. 19, no. 5, pp. 1–17, Sep. 2020.

Ms.G.YASASWINI is pursuing MCA from Audisankara college of Engineering &Technology (AUTONOMOUS), Gudur, Affiliated to JNTUA in 2024. Andhra Pradesh, India.

Author Profiles

Mrs. R. JYOSNA DEVI currently he is working Assistant Professor in Audisankara College of Engineering & Technology Gudur(M), Tirupati (DT), She is done MCA from Krishna Chaitanya institute of science and technology Nellore Post Graduation Centre in 2011,