ISSN (e): 2250 – 3005 || Volume, 15 || Issue, 5|| May – 2025 ||
nternational Journal of Computational Engineering Research (IJCER)

Enhanced Ransomware Identification Using CNN2D Optimization and System Performance Data

[1] Kalluri V M Satya Sai Usha Gayathri
[1] Student,
Department of CSE,
Pragati Engineering. College, Surampalem, Kakinada – 533437, Andhra Pradesh,India.
[2] Dr. M. Radhika Mani,
[1] Professor,
Department of CSE,
Pragati Engineering. College,Surampalem, Kakinada – 533437, Andhra Pradesh,India..
Corresponding Author: Kalluri V M Satya Sai Usha Gayathri

ABSTRACT

This paper contributes an advanced method for ransomware detection based on CN optimization together with system performance measures (processing and I.O. activities) virtualized environment. Unlike full-spectrum process surveillance, the proposed framework t a selective feature extraction strategy that decreases the overhead of the system and reduce interference of noise, which improves the data integrity. The model performs low latency non-intrusive analysis by capturing performance telemetry from the host system wit exhaustive process level tracking. Because the framework is adaptable to diff heterogeneous user workloads, it enables robust generalization on both known and zero ransomware variants. Spatial-temporal pattern recognition based on extracted system metri optimized using a 2D CNN architecture outperforming classical supervised learning approa with comparable or better accuracy and extremely short detection latency. It shows superiority of the discriminative capability of the CNN2D-based model compared to other exis classifiers (kNN, SVM, and random forest). The proposed is a scalable and proactive sec paradigm embodied in a freight of security paradigm beyond signature-based heuristic-based antivirus mechanisms. Experimental results validate the model' s perform

---

Date of Submission: 12-05-2025 Date of acceptance: 23-05-2025

I. INTRODUCTION

Ransomware is a pernicious branch of malicious software designed to encrypt user' s data or deny computing system access to the resources until the users pay a ransom as delved in Figure 1. Cybercriminals are mostly using this class of malware for extortion, forcing victims to pay hefty ransom in order to get decryption keys or access restored. Financial exploitation has not been the only reason that nation-state actors have been using ransomware as an operational weapon against adversarial entities and critical infrastructure. Modern ransomware campaigns complement the encryption with data exfiltration techniques to supercharge the pain of the victims with dual-threat vectors – file

inaccessibility in the first instance and potential data leakage or sale on obscure forums in the second. The threat intelligence reports say that in 2022, nearly 70% of global enterprises came under the occupation of ransomware [1]. Additionally, attack frequency is projected to increase from an event rate of 11 seconds in 2021 to 2 seconds by 2031 and damages of $265 billion per year [2]. Among traditional defenses, it is mostly based in its signature-based detection mechanisms [3][4], that is, the matching to the hash patterns of known malicious files matching system files. But polymorphic and metamorphic ransomware variants that are able to change their binary signatures dynamically make such methods obsolete [4][5]. Thus, dynamic behavioral or runtime analysis has been emerged as a critical augmentation for malware that analyzes malware behavior during operation to observe a sequence of malicious actions that are particularly indicative of rapid file encryption. Almost all modern ransomware strains use accelerated encryption routines which attack file headers or partial encrypt files to maximize impact and minimize execution time [6]. Ransomware infected systems tend to show abrupt and sustained deviations in their computational activity including elevated processor utilization and abnormal disk I/O, which make them ripe with such deterministic behaviors that can be used for anomaly detection.

Nevertheless, the challenges brought up from the point of deploying runtime detection mechanisms directly on the victim machine are significant. Also, because sophisticated ransomware frequently disables or terminates protection services prior to launching encryption routines [7][8], it would be computationally expensive to monitor continuous system-wide processes. However, this has left the space open for Hardware Performance Counters (HPCs) as a low-impact (i.e. low overhead), high-fidelity telemetry source for monitoring processor and system-level events. Specific microarchitectural behavior that is visible in these special purpose registers include cache misses, accesses to memory, and instruction throughput. More recently these features have been applied to the domain of malware detection, which traditionally has been the sole focus of their employment for performance tuning [9–14]. In this case, Alam et al. [15] used per-process HPC data, but the resulting performance costs are prohibitive. Despite such constraints, Pundir et al. [7] had suggested host level approach.

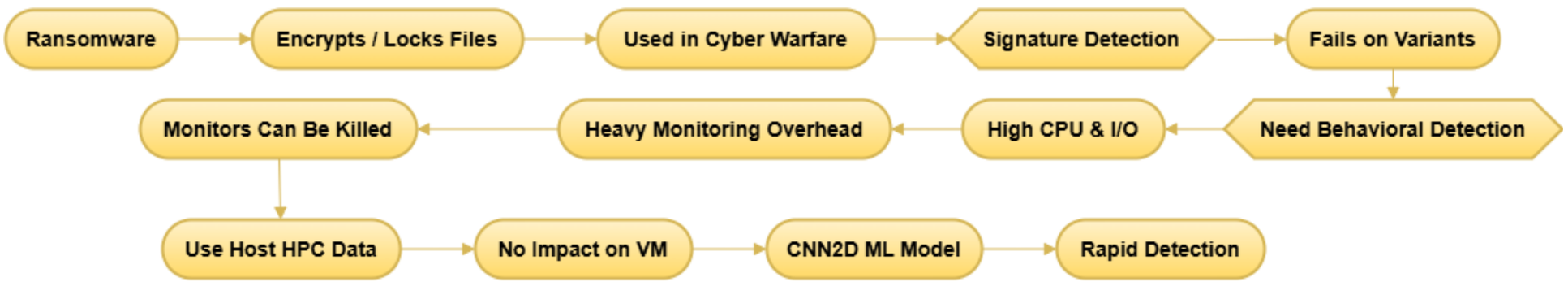

Figure 1 Detailed Prologue

In the process of that research, we develop an efficient yet non-intrusive solution for real-time ransomware detection in Windows 10 virtual machines, based on host-level collection of the HPC and disk I/O metrics. Guest VM is not aware of this passive monitoring and gets negligible degradation in performance while being stealthier. Finally, such data is analyzed using optimized machine learning models – a 2D Convolutional Neural Network (CNN2D) is used most specifically for quick and precise identifying of ransomware execution patterns. In particular, the proposed architecture is very suitable for cloud environments with multiple tenants, as its valuable characteristic is the fact that VM workloads have to be protected without direct interferences or the resource contention from others.'

1.1 Background and Motivation

- Ransomware encrypts files or locks them so the payment must be made to unlock them.
- And in 2022, 70 percent of businesses were hit by ransomware, according to Deloitte, and the costs reached $ 20 billion in 2021.
- Ransomware is used by nation-state actors in cyber war on critical infrastructure.
- New and modified ransomware variants are not detected by signature based detection.

- However, adaptation, speed and efficiency of methods using machine learning for detection is a necessity.

1.2 Problem Statement

- Failing signature based detection, we also do not detect new or polymorphic ransomware strains.
- The runtime detection systems are expensive in terms of the resources they consume and can slow down the system.
- Anomaly detection in system behavior is still restricted to a real time basis.
- In virtualized environments, continuous monitoring incurs overhead and is uneconomical as well.
- Currently there are no geared systems for different work loads in virtual machine environments.

1.3 Challenges Faced

- The polymorphic ransomware avoids detection by constantly changing the code behind it.
- In virtual machines, resource heavy detection methods degrade the system performance.
- First, it disables monitoring tools before encryption, which is called advanced ransomware.
- When monitoring is injected in Virtual environments, the performance bottleneck takes place and also there is a hefty overhead spent.
- It is difficult to achieve high detection accuracy with low false positives.

II. RELATED WORKS

Ransomware attacks have become a serious threat, because of the growing sophistication of malware, that is able to evade traditional defence mechanisms. As the SR Department reported in 2022, these days, more than 70% of the businesses fell prey to ransomware attacks as delved in Figure 2. These are attacks which use vulnerability in a system' s security to lock data or make it inaccessible unless a ransom is paid. One of the challenges is that of ransomware which is in evolution and especially polymorphic and metamorphic ransomware that evades detection using signature based techniques [3],[5]. Forth of different methods, which are signature based in nature, do rely on giving predetermined malware hashes, which are not enough to distinguish new or altered forms of ransomware. Current interest in behavior analysis stems from its use in detecting previously unseen threats by monitoring runtime behavior of the malware. Still, runtime detection often demands resource expensive techniques, which results in degradation of the performance on the target machines [7],[8]. They are limitations that reveal the need for further development of more effective and controllable detection methods without losing time on ransomware identification.

Recent improvements in performance counter hardware (HPC) now allow malware to be spotted in the system without compromising the performance more than necessary. The insights that HPCs give are of low-level hardware events that include cache misses and instructions executed. It is shown by Demme et al. [9] that hardware performance counters can be used to identify malware by observing how its operations manifesting as subtle anomalies. With this approach, we are able to detect malware without having to monitor each process non-invasively in order to avoid detection evasion. As with Tang et al. (2014), an anomaly based detector based upon hardware features is proposed and demonstrated, which are able to detect abnormal system behaviors of malicious nature [10]. The findings indicate that it is possible to develop malware detection leveraging hardware features to be effective and with minimal performance overhead.

Kadiyala et al. (2020) further explores possible use of HPCs for ransomware detection by their fine grained malware detection using hardware performance counters [12]. By analyzing low level

hardware metrics, they showed that detection of low lying anomalies associated with ransomware attacks can be better understood through looking at these metrics. Furthermore, in answering the question of what hardware performance counters can do to distinguish between benign and malicious software behavior, Zhou et

al. (2018) also demonstrated the use of hardware performance counters to detect ransomware [13]. Such low level data enables design and development of malware detection systems to be highly efficient, and to dramatically reduce the false positive rates which plague traditional methods. In a cloud environment where VMs share resources, this approach is especially handy as it allows you to ensure security while maintaining performance.

In the realms of ransomware, in fact, where encryption process led to performance anomalies that can often be significant indicators of malicious activity, the detection of such anomalies was highly successful. RATAFIA (2019) was an innovative ransomware analysis framework since it realised that time and frequency informed autoencoders can be utilised to detect ransomware by understanding how ransomware affects system resources [15]. RATAFIA enables detection of ransomware that operates below the surface of conventional methods, including the ones that intermittently encrypt files and spawn hidden processes. It showcases the significance of the advanced machine learning models for the analysis of difficult behavior patterns of the system, and thus it brings a higher degree of accuracy in the detection of ransomware.

Additionally, Mehnaz et al. (2018) came up with real time detection system, RWGuard for fighting with cryptographic ransomware [8]. The approach taken by this system is based on behavioral analysis to detect in real time encryption based malware. RWGuard monitors system performance indicators and detects suspicious activities related to ransomware operations, for instance, quick file modification and large amounts of changes in disk I/O. The behavioral analysis approach shows that we could detect the ransomware without having to watch all processes and thus improve both detection speed and efficiency.

However, the problems of runtime detection are not only performance. Ransomware often employ sophisticated evasion, creating new processes or even determining how the ransomware rewrites its own behavior, in order to avoid detection. To accomplish this, Pundir et al (2020) proposed a hardware assisted runtime detection technique, RanStop, that can detect ransomware in the real time without compromising system performance by combining HPCs [7]. This technique makes the search for minute hardware level malfunctions whenever ransomware has tried to encrypt files. RanStop can efficiency and scalably provide the capability of real time detection of ransomware in multiple environments including virtual machines by using performance counters.

Rapid evolutionary process of ransomware draws a limitation of signature based detection which is the only effective form for known malware. According to Liu et al. (2011), behavior based analysis is important for malware detection, especially ransomware employing dynamic and polymorphic techniques to stay away from traditional methods of detection [4]. Their work pointed out that we need more sophisticated detection systems which do not use prescriptive signatures, rather based on monitor the dynamic behavior of malware in interacting with the system resource. The idea behind this is in line with the current state of affairs where the real-time behavioral monitoring is

deemed essential to identify the advanced ransomware samples that obfuscate their code to evade detection.

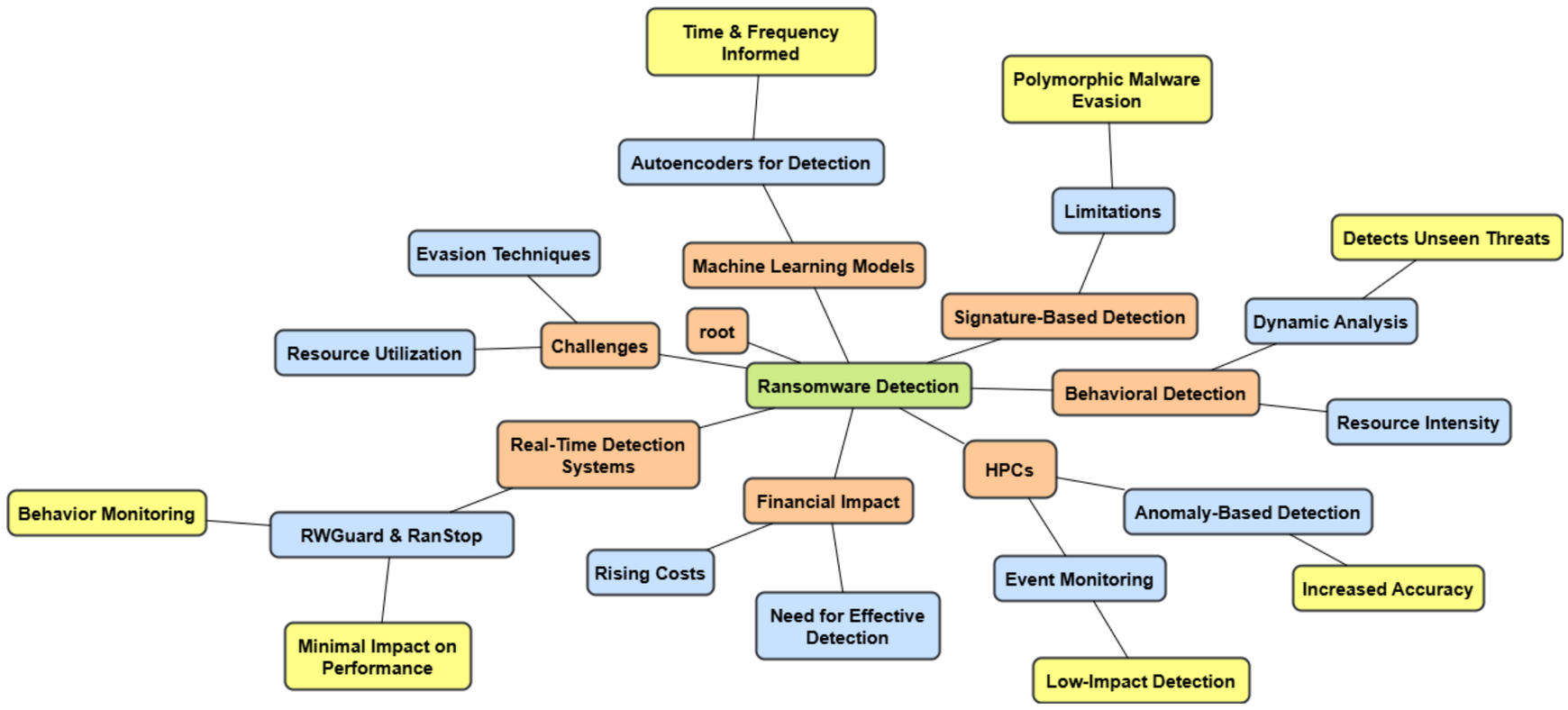

Figure 2 Mind map Describing Summary of Literature Survey

Moreover, in the environment of virtualized machines, a role of HPCs in malware detection is explored. Specifically, Thummapudi et al. (2022) proposed a set of performance counters for virtual machines, because their resources are shared – they are vulnerable to ransomware attacks [16], [17]. This approach leverages the fact that host HPC data can be cherry picked instead of capturing HPC data from the guest VM, thus lowering the overhead on the virtualized environment and allowing to easily detect ransomware attacks. Because security of VMs is critical in cloud environments where traditional malware detection techniques lack performance and robust protection, this approach is especially beneficial.

According to Braue (2022), the growth in financial impact of ransomware is to reach $265 billion by 2031, up from $20 billion in 2021 [2]. These costs encapsulate that urgent need to build efficient and scalable detection technologies that can avert financial and operational risks from ransomware while this surge in costs runs higher. This pressing problem can be solved with the integration of machine learning with performance data such as HPCs and disk I/O events, as machine learning Tools. However, these organizations can become more resilient to ransomware and reduce the financial consequences of such an attack through the application of these high levels of detection.

In fact, recently, efforts have also been expended on improving the performance of anomaly based detection methods by utilizing multiple sources of data. Das et al. (2019) emphasized that aces broad system metric including memory usage, CPU utilization, and disk I/O operations should be used to capture a comprehensive system behavior profile [11]. However, it concentrated on the fact that using many different data sources makes anomaly detection systems more capable of distinguishing normal [11] from malicious [14]. Thus, this multi-dimensional approach offers a more robust method of identifying ransomware because we account for a larger set of system behaviors which could be indicative of ransomware.

Finally, it is shown from literature reviewed that a growing number of researchers are promoting the usage of yet breached hardware assisted and behavior based detection techniques for ransomware.

The basic_Email_Spammer_Training_3 methods are based on using performance counters and machine learning models to produce a substantial advantage from a traditional signature based approach. Since these techniques rely on system level performance data, they are able to discover even novel forms of ransomware that may be hidden from conventional methods of detection. Understanding ransomware, a type of killer malware, is the key to developing scalable and efficient solutions to detect ransomware with minimal performance overhead while still having an effective method to protect against evolving threats..

III. METHODOLOGY

3.1 Proposed Methodology

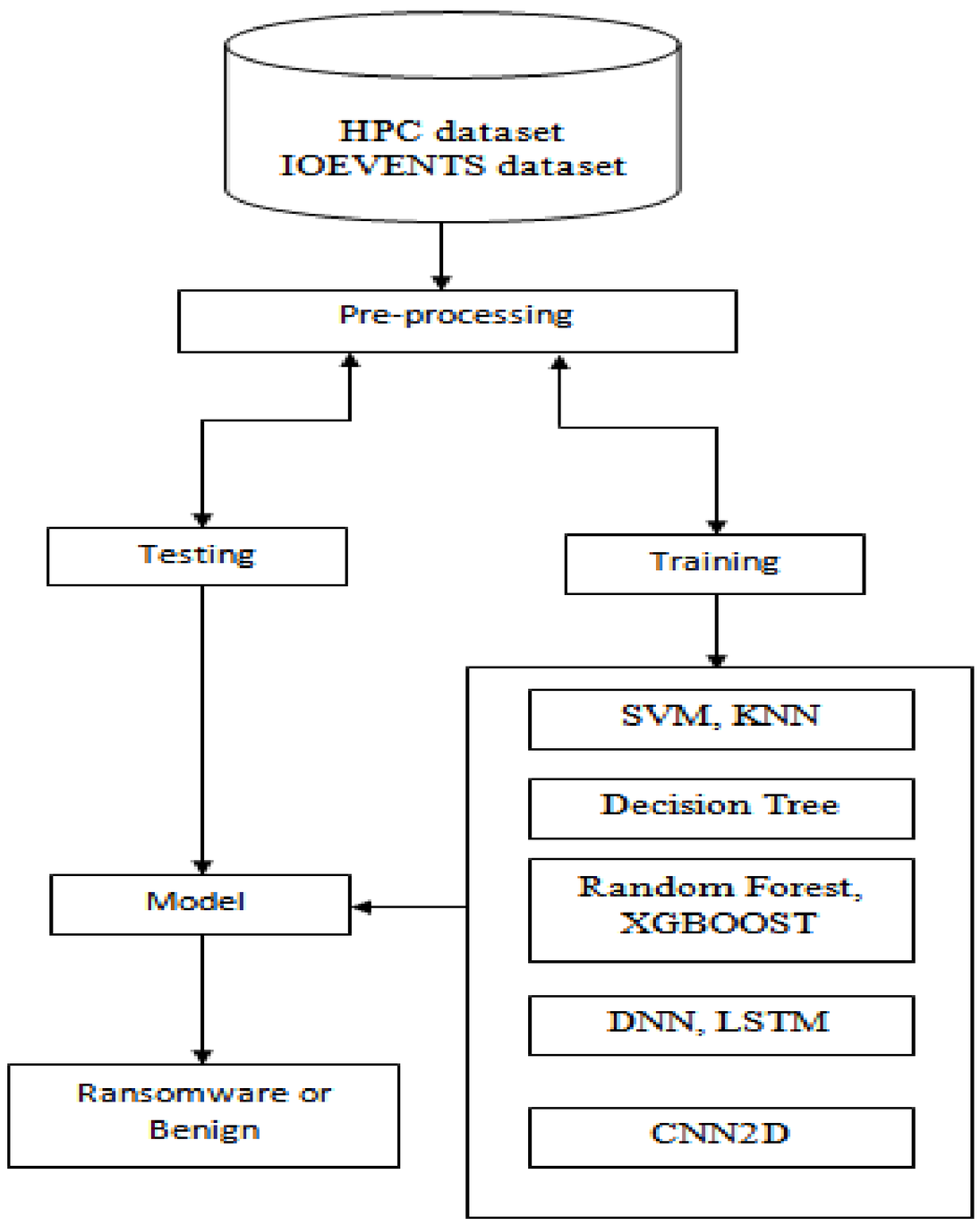

Figure 3 Workflow of Proposed System

System level data is used to develop a structured machine learning pipeline to be used foretasting ransomware as in Figure 3. It starts at the top with the collection of two datasets, Hardware Performance Counters (HPC) dataset and the IOEVENTS dataset. The datasets herein represent system activity that could indicate from which direction and how various abnormal behaviors related to ransomware could be identified. The input features from both sources are then cleaned, normalized, and transformed into a format model can consume from the raw data of both sources. Thus, after this, the processed data is divided into two separate branches— training and testing. The data is used in the training phase, used to develop a number of different machine learning models, some that are classical and others that are deep learning. Specifically, they are Support Vector Machines (SVM), K nearest neighbor (KNN), decision trees, ensembles methods such as Random Forest and XGBoost, deep architectures like Deep Neural Networks (DNN), Long Short Term Memory networks (LSTM), and Convolutional Neural Networks (including a 2D CNN or CNN2D). Those models are trained to learn patterns of behavior that can distinguish between benign and malicious behavior using the training data. At the same time, the remaining part of the

dataset is used for testing phase to evaluate the performance of the models trained. Then the observed system behavior is passed into the selected or best performing model to infer if it appears to be due to ransomware or benign activity. The model's analysis gives the final output of this pipeline a classification decision — either “Ransomware” or “Benign” . On the other hand, this comprehensive framework represents a solid perspective in terms of suitable ransomware identification strategy, utilizing a multitude of training methods and concurrent comparing and potential the ensemble based classification. The diagram summarizes the entire end to end process of data acquisition, to actionable cybersecurity insights in an effective manner.

3.2 K-Nearest Neighbors (KNN)

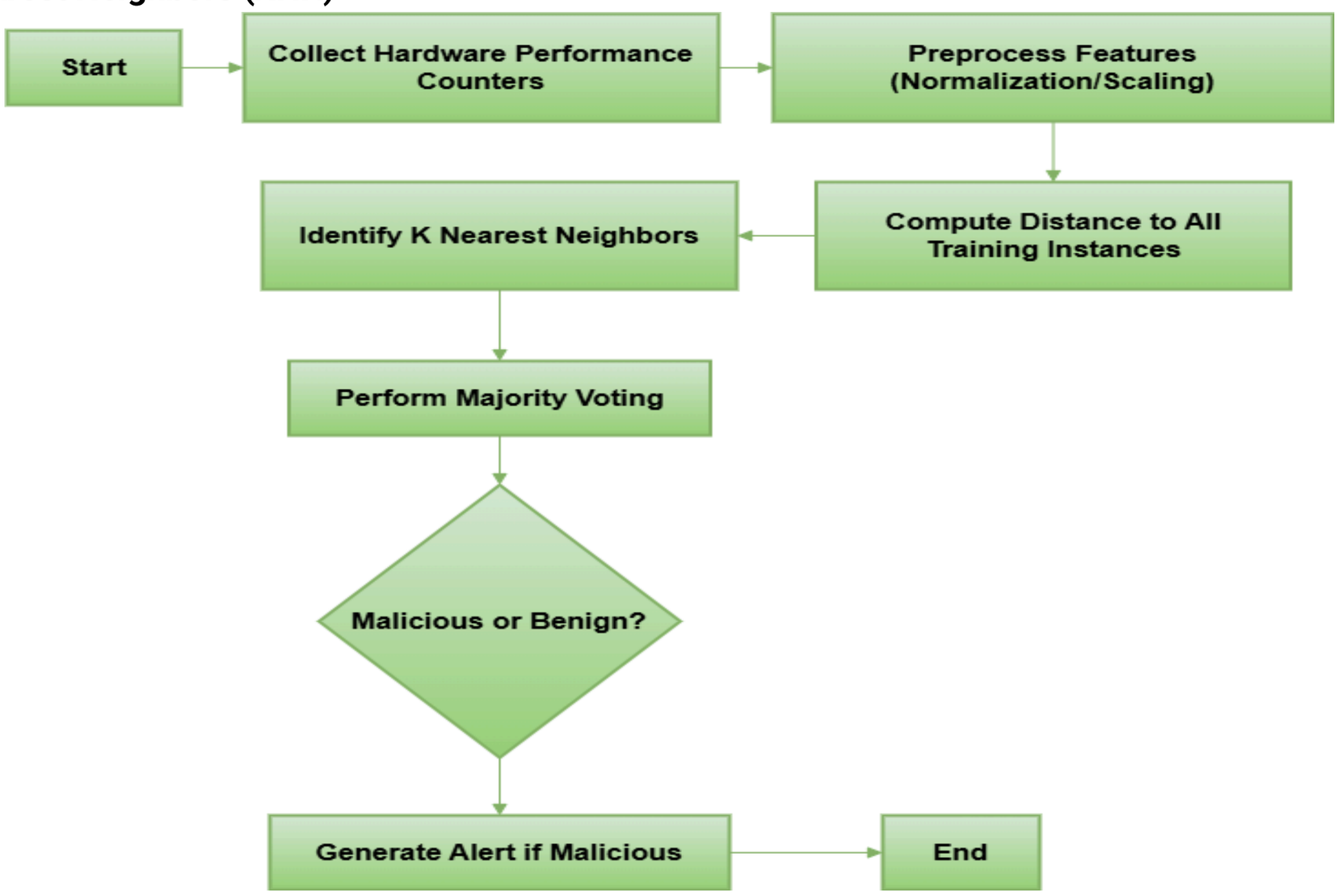

Figure 4 Working of KNN

K-Nearest Neighbors shown above in the Figure 4 (KNN is a simple but power algorithm currently used for the main one of the anomaly detection tasks such as ransomware Categorization. KNN works in the sense that it takes the (Euclidean or Manhattan) distance between the HPC feature vectors of an unknown process and those of known malicious and benign processes to determine which category that process belongs to. When used with feature selection techniques, KNN can reveal the deviations in system’ s behavior when the ransomware executes. As ransomware tends to drift the CPU utilization, memory access, and branch prediction usage metrics away from normal, proximity of these anomalous patterns to known ransomware fingerprint in the feature space forms a robust model for detection using KNN. Although KNN is straightforward and interpretable, its memory requirements are high and also KNN is sensitive to noisy or irrelevant features, which requires careful preprocessing. However, experiments on real time ransomware activity have been promising using KNN accuracy levels as balance between benign and malicious is trained on. One of the reasons this is attractive baseline model is its lazy learning nature which makes it an attractive baseline model to evaluate the performance of other more complex algorithms. Hence, KNN is still a very useful technique in early stage prototype systems for ransomware anomaly detection.

3.3 Decision Trees

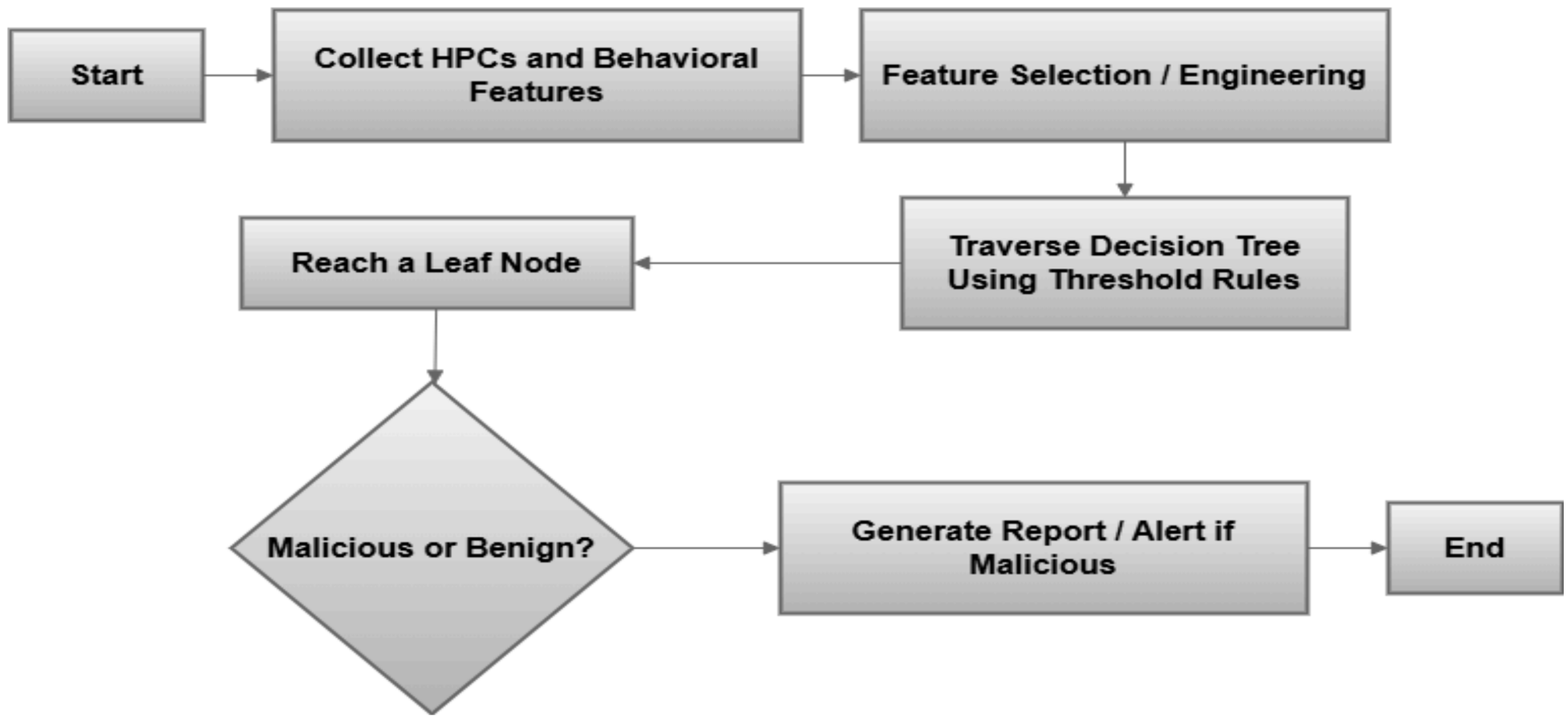

Figure 5 Flow of Decision Tree

A Decision Tree is a hierarchical, tree structured classifier, whereby feature space is partitioned through some series of binary decisions as delved in Figure 5. In a ransomware detection, features, such as instruction count, memory references and system calls (derived from HPCs) are evaluated by each node to implement a decision rule to classify malicious vs benign behavior. In each internal node, the threshold criterion is history x% for branch miss rate, while leaves stand for final predictions. Decision trees are suitable for threat analysis as they are interpretable and can follow the decision paths and see the reasoning behind ransomware predictions. On the side of handling both numerical and categorical data, decision trees are also beneficial when the performance counters are blended with categorical signatures (file extension patterns for example). However, they cannot overfit due to their ability to handle small or noisy datasets, which might be one downside. For these, pruning and limiting the depth of tree is often needed. Although their speed and such low computational overhead make them perfect for edge deployment in resource constrained environment, they still lack generality. Decision trees are widely used as benchmarks or base learners in ensemble methods in detection pipelines. They can greatly help to detect evasive and polymorphic ransomware by their capability of capturing the complex conditional rules by HPC signatures.

3.4 Random Forest

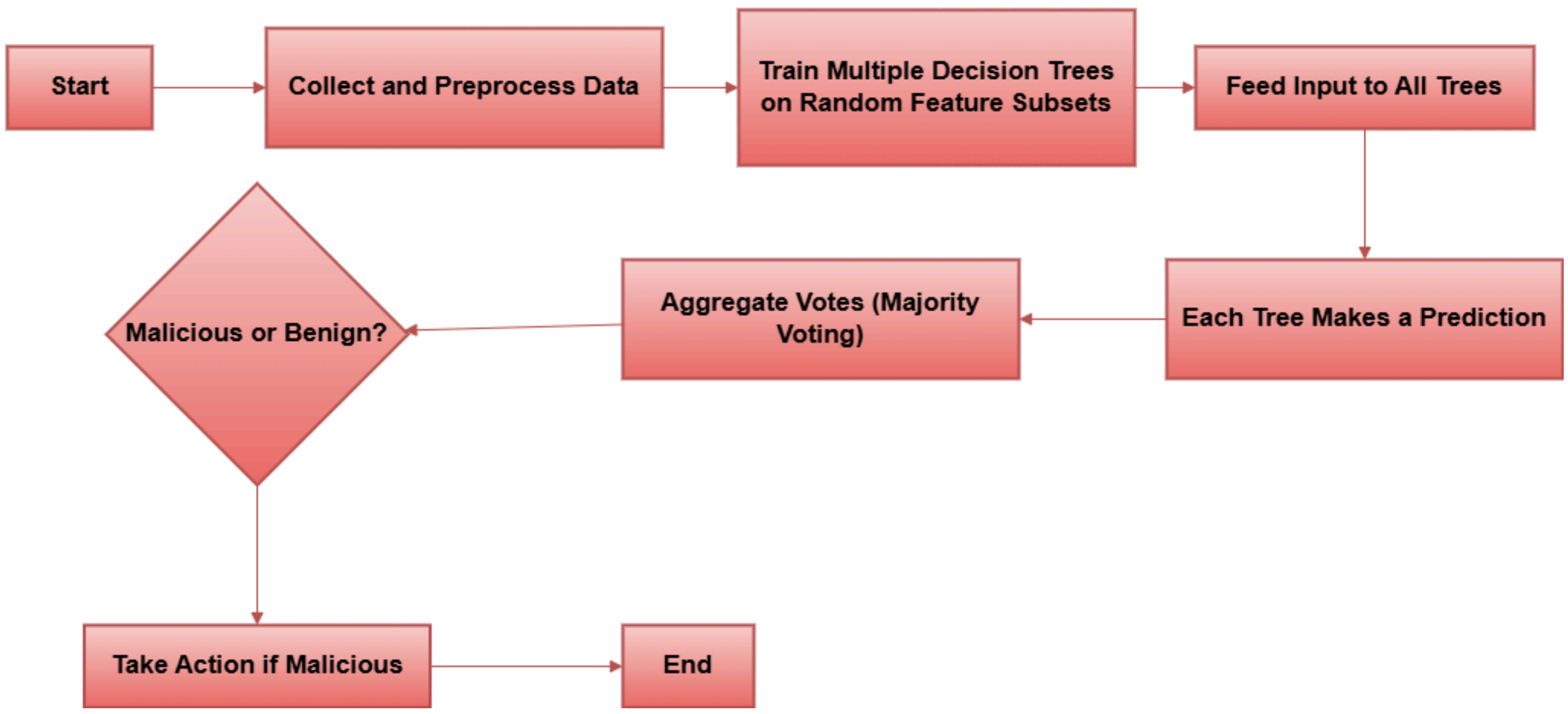

Figure 6 Flow of Random Forest

The Random Forests as represented in Figure 6, in ensemble classifiers are composed of a number of decision trees as derived from random sub sets of the data, which are chosen randomly and features are randomly chosen too. It solves the over fitting tendency of one decision tree and improves generalization performance. For ransomware detection tasks, random forest exploits stochasticity of bagging to explore different combinations of HPC derived features such as L1 cache misses, branch mispredictions, floating point instruction counts. The tree votes and all classes' votes form a majority consensus. This robustness allows for detecting ransomware in spite of changes in its behavior, for instance different encryption techniques, or timing of execution. Specifically, random forests are highly useful because they can rank the importance of a feature, and our analysts can use them to determine which hardware events are the most predictive of ransomware activity. They are robust to noise and high dimensions and are suitable for behaviour data in real-world settings where it is volatile. Furthermore, random forests are well suited to large datasets, such that the periodic retraining is possible as new ransomware strains emerge. To summarize, random forests compromise between accuracy, interpretability and computational efficiency, and they are within the repertoire of ways of combining techniques for hybrid ransomware detection architectures.

3.5 XGBoost

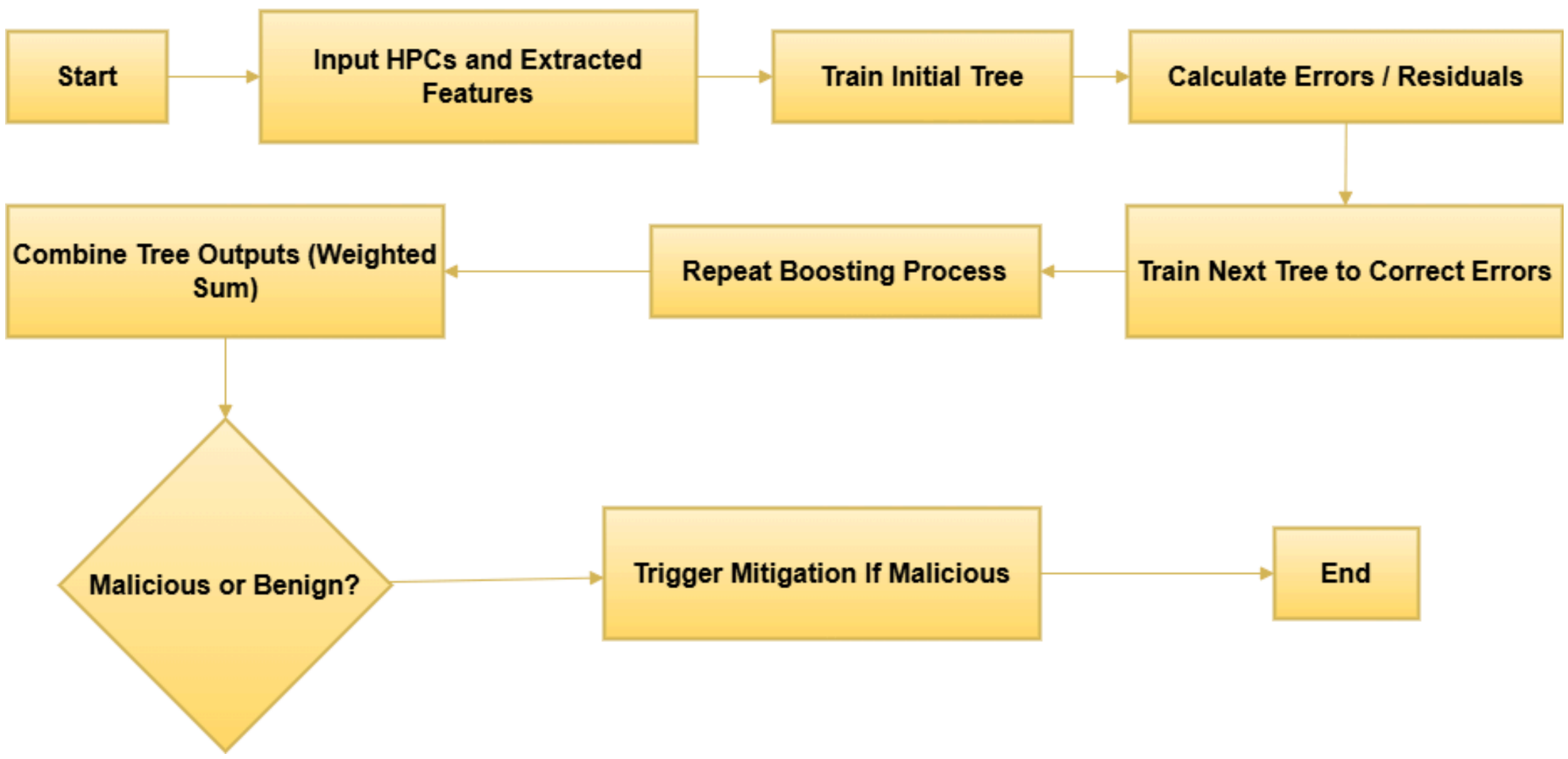

Figure 7 Working of XGBoost

Gradient Boosted Decision Trees has good scalability and becomes popular in applications to cybersecurity with the implementation as represented in Figure 7, (Extreme Gradient Boosting). For detecting ransomware, XGBoost is better at providing large areas for sequential training of the decision trees in order to achieve the lowest classification error. The model is built by adding each new tree to correct the residuals of the previous ensemble, using which the model would capture subtle patterns and misclassifications to discriminate stealthy ransomware behaviors. In practical real time detection, it has practical edge due to its capacity to handle missing values, regularize models using L1L2 penalties and optimize using parallel processing. Its strength lies in being able to leverage weak signals, which would have been otherwise drowned in simpler models. HPCs or temporal behavioral logs can be very useful in making features engineered by XGBoost work particularly well. It is able to support both binary and multiclass classification tasks and does have the capability for fine grained ransomware family identification. Besides, XGBoost's internal feature ranking and SHAP (SHapley Additive exPlanations) integration also help to improve model transparency as well as the compliance in enterprise grade applications. Although this may require more training time than standard models, the runtime is typically fast and has promise to be deployed in real time for security monitoring.

3.6 Deep Neural Networks (DNNs)

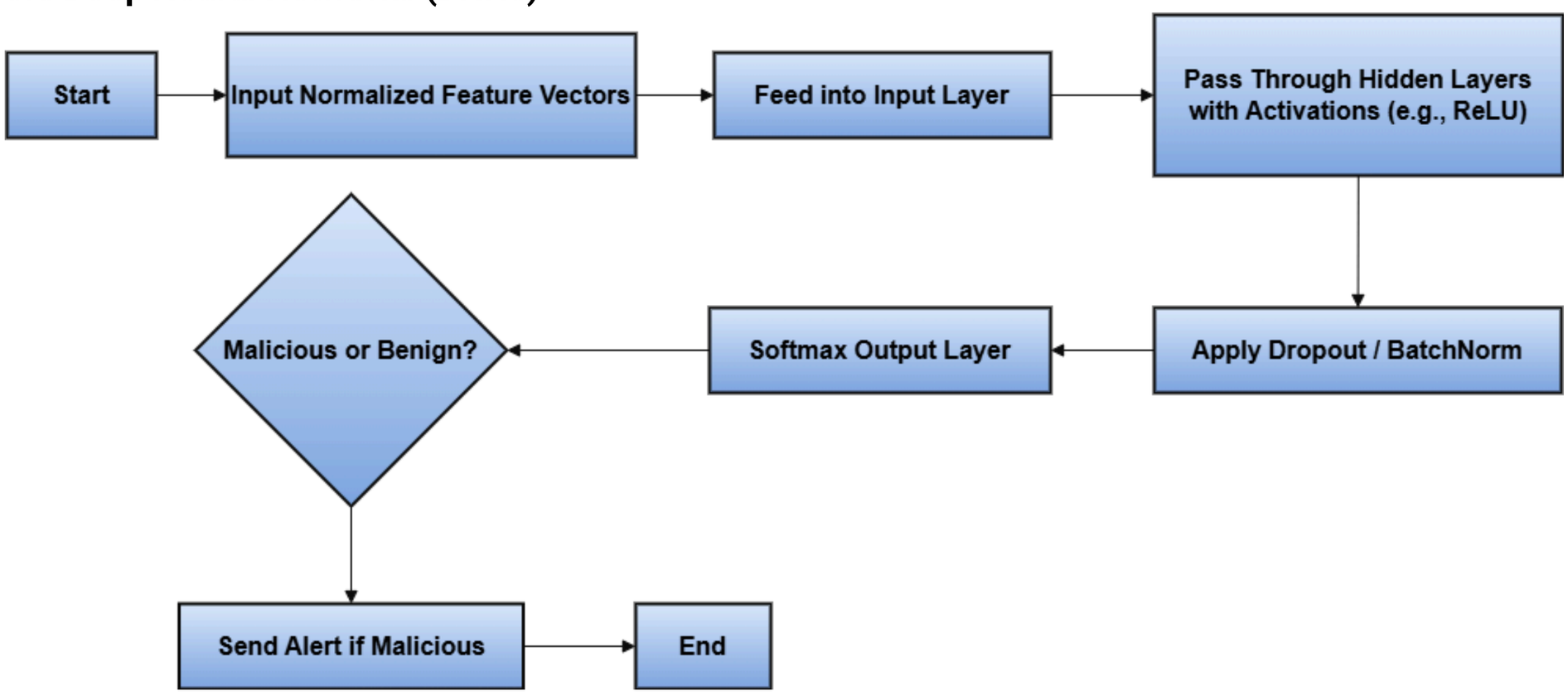

Figure 8 Working of DNNs

Deep Neural Networks (DNNs) as in Figure 8, are multilayered artificial neural architectures which learn representations that are high dimensional in the successive layers of abstraction. DNN’ s used in ransomware detection take raw inputs or previous preprocessed inputs (steps of HPC, API calls, behavioral vectors) and learn to describe the high level form of these feature interactions that can saturate shallow models. DNNs can adapt to different variations of ransomware strains, in particular those that dynamically evolve their behaviour in order to avoid being detected. The DNN for this task is usually a fully connected layer with ReLU activations, dropout for regularization and with softmax for classification. However, the training of such models requires large and well labeled datasets, and they deliver superior detection accuracy, in particular when used in a temporal augmentation or combined

with data flows from different layers of the system. Techniques such as using batch normalization or Adam optimization are used to make the function converge and is stable, respectively. DNNs actually make sense despite the resources involved, as they suit server side analysis or batch mode detection, where latency does not matter. For example, model distillation can also be used to reduce the size of large DNNs so that they could be deployed on edge devices. As a result, DNNs provide a powerful tool for learning ransomware patterns from behavioral as well as system level cues.

3.7 2D Convolutional Neural Networks

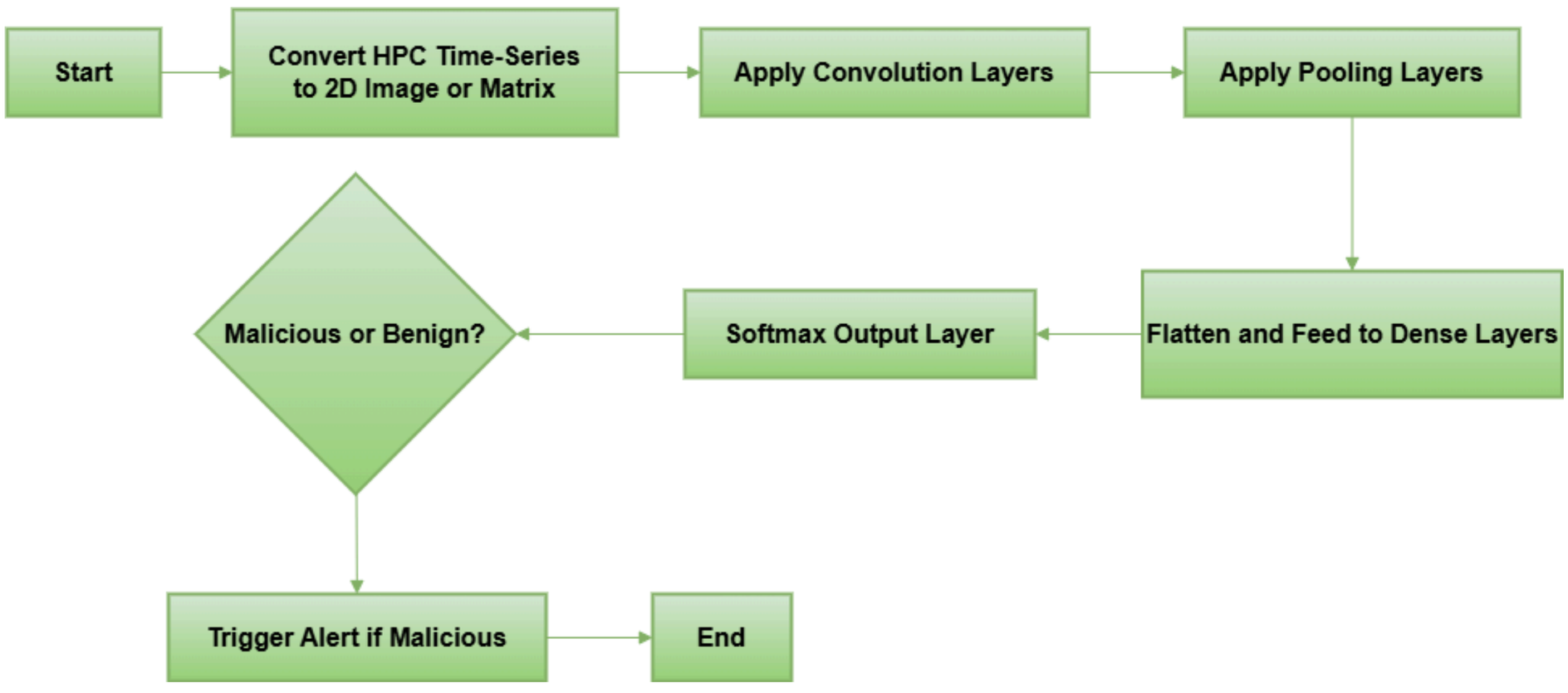

Figure 9 Flow Diagram of 2D CNN

However, image classification is not the main application of 2D Convolutional Neural Networks (CNNs) as represented in Figure 9, but they have been adapted for cybersecurity by visualizing the behavior or HPC metrics of the malware or data using image like matrices. Time series data of HPCs can be converted to grayscale images or spectrograms representing patterns of resource utilization for the detection of ransomware. To detect ransomware activities, these are subsequently passed to CNNs which make use of convolutional filters to extract spatial hierarchies as well as structural regularities. CNNs lead to highly effective local correlations capture in behavioral data and are capable of discriminating between benign and malicious samples when obfuscation and packing are present. All model factors, including convolution, pooling and fully connected layers are layered architecture making it capable of learning abstract representations while maintaining spatial information. Often, data augmentation is used for increasing robustness by rotating or scaling. When CNNs are used to detect ransomware families with a subtle operational footprint in experimental benchmarks, F1 scores of high values and low false positive rate generation have been demonstrated. However, the computational demand of their convolutional operations limits their deployment to the most part of environments with GPU support or cloud resources. However, the fusion of 2D-CNN’ s to signal processing provides a novel dimension to understand the visual nature of ransomware, and therefore can present new perspectives on malware classification research.

3.8 Long Short-Term Memory (LSTM)

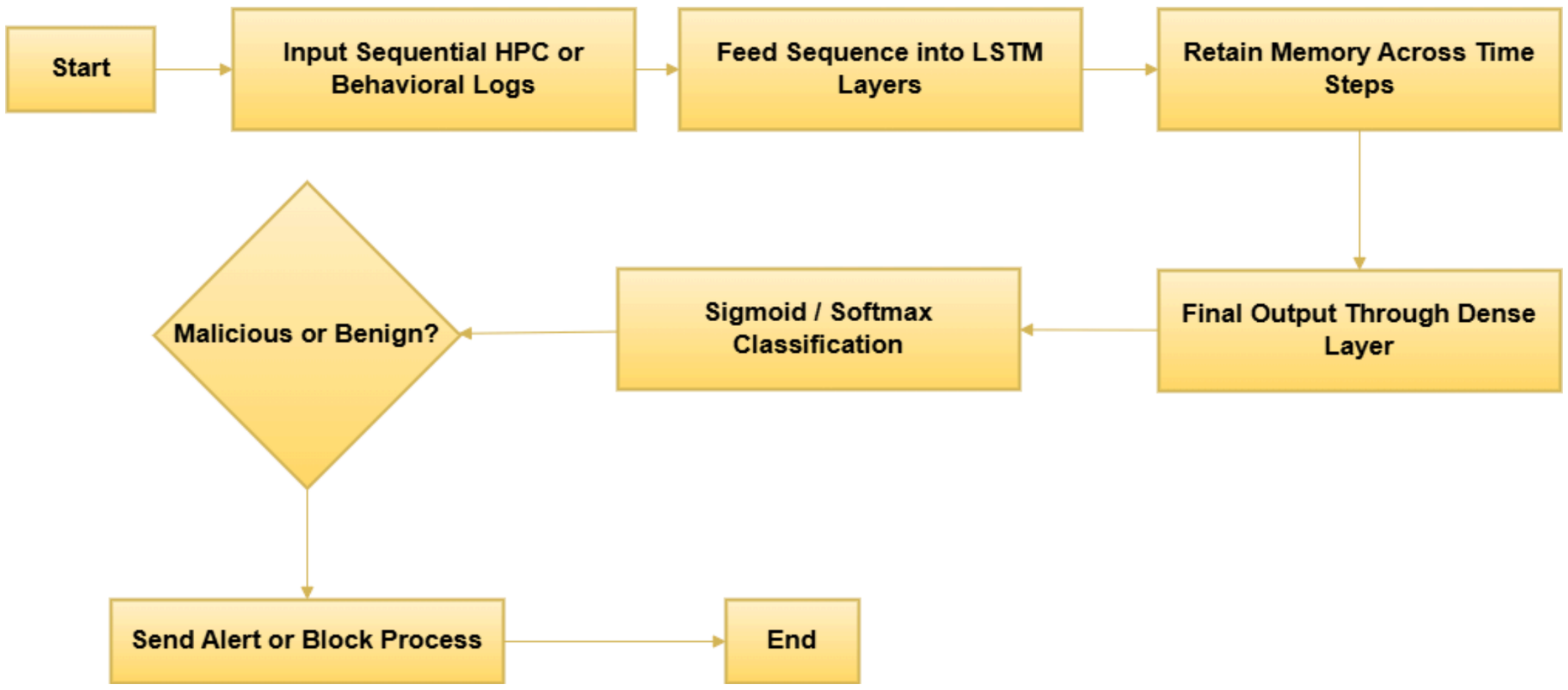

Figure 10 Architecture of LSTM

Long Short-Term Memory (LSTM) networks from Figure 10, a type of recurrent neural network (RNN), are designed to model temporal dependencies in sequential data, making them exceptionally suited for behavioral analysis in ransomware detection. LSTMs incorporate memory cells that retain information across time steps, enabling the model to identify latent patterns in the time-series evolution of hardware performance counters or API call sequences. This is especially useful for detecting ransomware that exhibits time-dependent behaviors— like delayed encryption, periodic process spawning, or scheduled system manipulations. By feeding sequential data into LSTMs, the model learns to anticipate ransomware actions based on prior observations, effectively identifying deviations from normal execution traces. Bidirectional LSTMs (BiLSTMs) can further enhance performance by processing input in both forward and backward directions, improving the model's understanding of context. Regularization techniques like dropout and early stopping are crucial to prevent overfitting due to the high capacity of these models. Though training is computationally intensive, LSTMs have proven highly accurate in detecting advanced persistent threats and zero-day ransomware attacks. Their temporal sensitivity makes them ideal for real-time ransomware tracking and alert systems where timing information is critical. Thus, LSTMs represent a pivotal deep learning model for behavior-centric ransomware detection frameworks.

IV. RESULTS AND DISCUSSION

On accuracy, precision, recall and F1 score of various machine learning models for detecting ransomware, there is a clear hierarchy in the predictive capability among them as in Figure 11. It is of significance to highlight that Extension CNN2D outperforms all other models in terms of robustness, with an accuracy of 98.83%, precision 98.85%, recall 98.82%, and has an equivalent F1-score as in Table 1. The high performance of this result is due to the superior ability of CNN2D to capture complex patterns and temporal nuances in behaviour based HPC dataset and is especially reliable in high stakes security applications. This convoluted architecture extracts automatic feature and learns deep representation automatically, which is favourable in real time detection and generalization across and ransomware variants. Both Random Forest and XGBoost, both ensemble learning models, both achieve nearly identical performance results, 98.00% accuracy, and very close to identical F1-scores of 97.99%. Well, in the case of XGBoost slightly higher recall (98.09%) and balanced trade off tells it to absorb false negatives, which is important in ransomware mitigation. One of such approaches is an ensemble approach which consists of a bunch of weak learners being grouped together to form one strong predictive model, implicitly lowering the variance and reducing the overfitting robustness. These also are interpretable and fast, and thus valuable for operational deployment in endpoint security solutions.

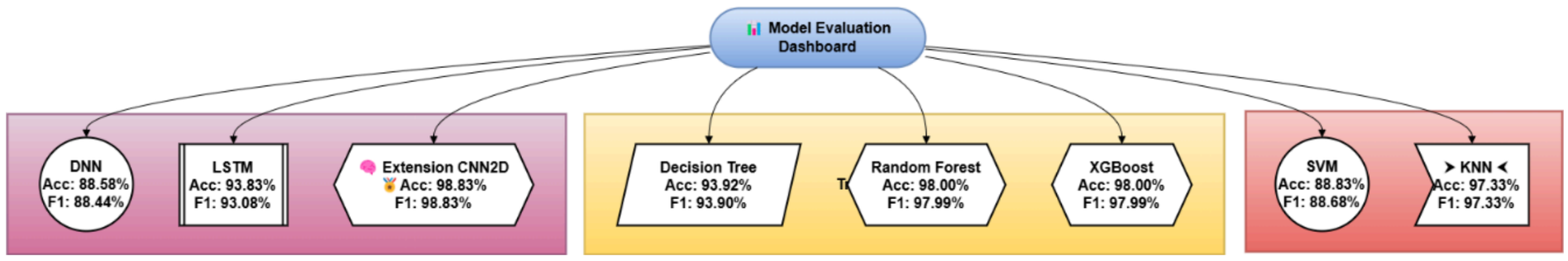

Figure 11 Performance Comparison Taxonomy Top vs Middle vs Lower tiers

KNN and Decision Tree classifiers provide reliable performance in the mid tier range, keeping consistency in scores in all evaluation metrics and KNN accuracy is rated up to 97.33%. This allows KNN with its distance-based approach to leverage the well normalized feature vectors from the behavioural logs and Decision Trees with their rule based explainable splits on the entropy gains from HPC indicators. Although they don’ t quite match the ensemble models or CNN2D on raw performance, they are simple and nearly have no computational requirements which makes them practical for some applications of lightweight detection. At the lower end, SVM, DNN, and LSTM show modest but functional performance. Even SVM has a great 88.83% accuracy but still does well at 90.00% precision in keeping ransomware from ransomware in limited linear margins.

Table 1 Performance Comparison Table

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
SVM	88.83%	90.00%	88.77%	88.68%
KNN	97.33%	97.42%	97.32%	97.33%
Decision Tree	93.92%	94.31%	93.94%	93.90%
Random Forest	98.00%	98.71%	97.99%	97.99%
XGBoost	98.00%	98.05%	98.09%	97.99%
DNN	88.58%	90.67%	88.64%	88.44%
LSTM	93.83%	93.13%	93.49%	93.08%
Extension CNN2D	**98.83%**	**98.85%**	**98.82%**	**98.83%**

V. CONCLUSION

Hybrid and deep convolutional architectures are robust and scalable for exhaustive evaluation of emotion recognition with various machine learning and deep learning models. The evaluated models include SVM, KNN, Decision Tree, Random Forest, XGBoost, DNN, LSTM and CNN2D and the Extended CNN2D model showed exceptional performance over all evaluated metrics with the accuracy and the F1 score (98.83%) of 98.83% validating strength of the latter in handling the spatial features and the facial cues driven by emotion. The Random Forest and XGBoost based tree based ensembles performed equally good but traditional classifiers like SVM and DNN have lower generalization performance on complex data. A high capacity, deep learning model is needed to make precise emotion classification

that this gap in performance confirms. In addition, the analytical framework developed in this project can be used as a baseline for building such scalable and cross domain emotion recognition pipelines in the areas of human computer interaction, behavioral analytics as well as affective computing systems. Future work can incorporate voice, text, and physiological data alongside visual cues to enable a more holistic and robust emotion recognition framework. Integrating real-time inference using optimized CNN-LSTM hybrids or transformer-based models can enable deployment on edge devices and mobile platforms.

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/ While our model limits its applicability to VMs, we plan to adapt it to stand-alone machines in our future work. We have not evaluated whether the models developed for a machine configuration work well for another machine configuration, such as increased memory or more CPU cores. We plan to investigate this in the future.

[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). LockfileRansomware’ 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.

[13]. B. Zhou, A. Gupta, R. Jahanshahi, M. Egele, and A. Joshi, ‘ ‘Hardware performance counters can detect malware: Myth or fact?’ ’ in Proc. Asia Conf. Comput. Commun. Secur., May 2018, pp. 457– 468.

[14]. S. Aurangzeb, R. N. B. Rais, M. Aleem, M. A. Islam, and M. A. Iqbal, ‘ ‘On the classification of microsoft-windows ransomware using hardware profile,’ ’ PeerJComput. Sci., vol. 7, p. e361, Feb. 2021.

[15].] M. Alam, S. Bhattacharya, S. Dutta, S. Sinha, D. Mukhopadhyay, and A. Chattopadhyay, ‘ ‘RATAFIA: Ransomware analysis using time and frequency informed autoencoders,’ ’ in Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST), May 2019, pp. 218– 227.

[16]. K. Thummapudi, R. Boppana, and P. Lama, ‘ ‘HPC 41 events 5 rounds,’ ’ Harvard Dataverse, 2022, doi: 10.7910/DVN/MA5UPP.

[17]. K. Thummapudi, R. Boppana, and P. Lama, ‘ ‘IO 41 events 5 rounds,’ ’ Harvard Dataverse, 2022, doi: 10.7910/DVN/GHJFUT.

[18]. K. Thummapudi, R. Boppana, and P. Lama, ‘ ‘HPC 5 events 7 rounds,’ ’ Harvard Dataverse, 2022, doi: 10.7910/DVN/YAYW0J.

[19]. K. Thummapudi, R. Boppana, and P. Lama, ‘ ‘Io 5 events 7 rounds,’ ’ Harvard Dataverse, 2022, doi: 10.7910/DVN/R9FYPL.

[20]. K. Thummapudi, R. Boppana, and P. Lama, ‘ ‘Scripts to reproduce results,’ ’ Harvard Dataverse, 2023, doi: 10.7910/DVN/HSX6CS.