Hybrid Machine Learning Approach for Early Detection of Ransomware Behavior

Submitted By

Arafat Hossain Pranto
(212-35-3178)
Department of Software Engineering

Supervised By

Ms. Nadira Islam
Assistant Professor
Department of Software Engineering

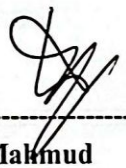
A thesis submitted as a partial requirement for the completion of the Bachelor of Science degree in Software Engineering.

Fall 2025

APPROVAL

This thesis titled on **"Hybrid Machine Learning Approach for Early Detection of Ransomware Behavior"**, submitted by **Arafat Hossain Pranto (ID: 212-35-3178)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



Chairman

Dr. S M Hasan Mahmud
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Tapushe Rabaya Toma
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Khalid Been Badruzzaman Biplob
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External Examiner

Dr. Md. Sazzadur Rahman
Professor
Institute of Information Technology
Jahangirnagar University

DECLERATION

I hereby declare that the work presented in this thesis is my own original research work, carried out under the supervision of **Ms. Nadira Islam**, Assistant Professor, Department of Software Engineering, Daffodil International University.

This work has not been submitted anywhere, either in whole or in part, for any degree, diploma, or publication in this or any other university. All sources of information used in this thesis have been duly acknowledged.

Supervised By

Ms. Nadira Islam
Assistant Professor
Department of Software Engineering

Submitted By

Arafat Hossain Pranto
ID: 212-35-3178
Department of Software Engineering

ACKNOWLEDGEMENT

First and foremost, I want to thank God for giving me the strength, patience, and willpower to do this research task effectively. This achievement would not have been achievable without His blessings. I am really grateful to my wonderful parents for always being there for me, encouraging me, and giving me advice during my school years. Their sacrifices and faith in me have always inspired me. I want to thank **Prof. Dr. Imran Mahmud**, the Head of the Department of Software Engineering, for creating a friendly academic atmosphere and giving the resources I needed to do my job. I want to thank my supervisor**, Ms. Nadira Islam**, from the bottom of my heart for all the help, advice, and feedback she gave me over the whole study process. His knowledge and support have been very important in the development of my thesis. I also want to thank all the professors in the Department of Software Engineering for their commitment to academic excellence and for teaching me the skills and knowledge I needed to do this project well.

Lastly, I want to thank my friends, classmates, and coworkers at Daffodil International University (DIU) for their help, encouragement, and constructive conversations during this endeavor.

ABSTRACT

The pervasive and escalating threat of ransomware necessitates the development of extremely flexible and robust detection systems that exceed the restrictions of present signature-based and heuristic techniques. This thesis introduces a novel hybrid machine learning framework for the early identification of ransomware activity, specifically exploiting Application Programming Interface (API) call patterns. The main new idea is a smart way to set the weights for Recurrent Neural Networks (RNNs) at the beginning. This is done by using coefficients from a pre-trained Logistic Regression model as the initial input-to-hidden weights. This strategic integration tries to address common difficulties in neural network training, such as vanishing/exploding gradients and sluggish convergence, while enhancing overall detection accuracy and efficiency.

The recommended model was thoroughly examined on a complete dataset comprising both malicious and benign API call sequences, assessing its performance across both balanced and imbalanced data distributions. Experimental results demonstrate the greater efficacy of the hybrid technique. On a balanced dataset, the model acquired an accuracy of 83% with a greatly reduced optimal loss value of 0.44, outperforming a baseline RNN seeded with Xavier weights (loss of 3.47). When analyzed on an imbalanced dataset, which more closely matches real-world settings, the hybrid model attained an incredible 98% accuracy, topping the Xavier baseline's 92%, while preserving comparable low loss levels. These findings underline the model's robustness and its capacity to generalize effectively across different data aspects. This research contributes a valuable viewpoint on boosting neural network training by intelligent weight initialization, giving a more resilient and efficient strategy for tackling the dynamic environment of ransomware attacks in cybersecurity.

Table of Contents

Chapter 1

Introduction

1.1 Background and Evolution of Cybersecurity Threats

The cybersecurity landscape has experienced a significant transition, characterized by an ongoing increase in the number and complexity of cyberattacks. This requires the ongoing advancement of innovative methods for identifying and mitigating harmful actions. [1], [2]. Statistical data highlights the severity of this issue: by the end of 2021, more than 1.2 billion malware samples had been documented, with a frightening average of 350,000 new copies found each day. This extraordinary growth rate implies not only a numerical surge; it indicates a fundamental transformation towards a highly industrialized, professionalized, and quickly developing cybercrime ecosystem. The fundamental adaptability and sheer magnitude of these emerging threats render conventional, reactive, signature-based defenses progressively unsuitable. Cybercriminals regularly upgrade their strategies and ways to avoid recognized detection technologies. For instance, targeted attacks showed a 42% surge in 2021, ransomware becoming much more disruptive and financially rewarding. Simultaneously, supply attacks soared by 93%. The advances underscore an urgent necessity for creative, intelligent, and automated solutions to successfully control the everexpanding malware sector. This escalation necessitates a strategy move towards predictive and behavioral models, moving beyond just reactive approaches. The massive influx of new malware samples makes it impossible for human analysts or static signature updates to keep pace, rendering automation and machine learning not just advantageous but critical for sustaining effective cybersecurity defenses.

The foundation of cybersecurity is malware analysis, which calls for the methodical examination, comprehension, and disassembly of dangerous software. This method provides crucial information about the purpose, functionality, and anticipated impact of malware on digital ecosystems [3].

1.2 The Escalating Threat of Ransomware

Among the myriads of cyber dangers, ransomware has emerged as one of the most important and damaging kinds of cybercrime. This malicious software is specifically designed to encrypt a victim's data, rendering them unavailable, and then demands a ransom payment for their release. The implications of ransomware attacks are significant, ranging from substantial financial expenditures to the full paralysis of activity across important domains such as healthcare, finance, and government. Over the past decade, ransomware has witnessed tremendous evolution, with attackers regularly improving their plans and broadening their tactics to defeat existing protective solutions [4].

The increased sophistication and frequency of these attacks have rendered conventional detection methods more ineffectual. Signature-based solutions, for instance, rely on known characteristics of malware, rendering them inherently vulnerable to new, previously unreported strains that employ polymorphic methods, encryption, and other evasion strategies. Conversely, heuristic-based methods, while seeking to detect dangerous activity by monitoring program operations, generally produce a high proportion of false positives, particularly in complicated or dynamic computing scenarios. These constraints underline an urgent need for inventive detection systems that do not rely only on static signatures or known patterns.

Ransomware typically targets Windows systems, creating massive disruption by modifying files. The primary method of ransomware is file encryption and manipulation, which manifests as a specific, observable series of behaviors on the file system. These characteristics include the rapid generation, deletion, renaming, and modification of files. This inherent behavioral footprint, unlike static signatures, is incredibly difficult for attackers to totally hide without undermining the ransomware's essential operation. Because of this, behavioral analysis—especially of file system activity and API calls—is a very pertinent and promising detection method. Early identification is essential due to the destructive nature of ransomware, which is characterized by data encryption and disruption of operations. The harm has already been done if discovery doesn't happen until after the encryption procedure is finished. Finding the beginning or early stages of these file system modifications or API call sequences is therefore essential, and it serves as the foundation for this thesis' focus on "Early Detection of Ransomware Behavior [5]."

1.3 Role of Behavioral Analysis and Machine Learning in Cybersecurity

Malware analysis remains a crucial field of cybersecurity, involving the rigorous research of malicious software to understand its behavior and impact on systems. inside this domain, dynamic analysis, which entails observing malware actions inside a controlled environment, is particularly beneficial against zero-day and camouflaged threats. By investigating trends in API call sequences, these algorithms can infer destructive operations performed by the infection, delivering a robust technique for detection.

Machine learning algorithms offer a promising route for uncovering unknown ransomware variations by learning from patterns hidden inside big datasets. These techniques are successful in recognizing abnormalities in system behavior that may indicate the existence of ransomware [6]. Deep learning models further increase detection capabilities by automatically extracting nuanced features from raw data, enabling them to catch sophisticated and subtle patterns that can evade earlier methods. The integration of machine learning with conventional detection methodologies can produce a more complete and resilient solution, capable of adapting to the ever-evolving threat landscape.

1.4 Challenges in Advanced Malware and Ransomware Detection

Despite progress, implementing complex malware and ransomware detection systems still confronts considerable obstacles. Because they rely on a database of known signatures, traditional signature-based approaches are inherently susceptible to novel, hitherto undisclosed malware strains [7]. Similar to this, heuristic-based systems can have significant false-positive rates even when they strive to identify questionable activities. This might diminish operational effectiveness and generate alarm fatigue among security staff. The ongoing innovation of attackers, who use complex evasion measures including encryption, packing, and polymorphism techniques to mask their harmful code and behavior, is a major contributing cause to the constraints of these classic methodologies.

While machine learning (ML) offers great capabilities, its application in cybersecurity is not without difficulties. ML models usually demand high-quality labeled data for training, which becomes incredibly difficult to get for zero-day or novel ransomware outbreaks that have not yet been detected or categorized. Furthermore, a sole dependence on machine learning can lead to concerns such as overfitting, where the model performs well on training data but fails to generalize effectively to new, undiscovered attack types. A

basic problem inside neural networks, a subtype of deep learning, resides in weight initialization. The base study specifically mentions this as a problem that its proposed hybrid approach intends to remedy.

Moreover, real-world cybersecurity datasets are intrinsically skewed, meaning that harmful samples are substantially uncommon than benign ones. For instance, the dataset utilized in the fundamental study for this thesis demonstrated an imbalance where the malware category was 40% greater than the benign category [8]. Such dataset imbalances might result in substantial performance issues, potentially biasing the model towards the majority class and resulting to inferior detection for the minority class. This underlines that the constraints of ML models are not only technological difficulties but basic data challenges. The necessity for thorough data curation and preprocessing, including approaches like undersampling to balance datasets, is a significant practical factor that directly influences the reliability and deployability of detection systems in real-world applications. Effective ransomware detection is not just about selecting the proper algorithm but also about tackling certain underlying data challenges to ensure the reliability and generalizability of the solution.

1.5 Contributions of this Thesis

This thesis provides several significant advances to the world of cybersecurity and machine learning, specifically in the realm of ransomware detection:

- **Novel Hybrid Model Adaptation:** This research adapts and greatly expands upon the innovative hybrid Logistic Regression and Recurrent Neural Network (LR-RNN) model, specifically utilizing it for the early identification of ransomware activity. This proves the model's applicability and performance beyond broad malware detection to a highly critical and specialized threat.
- **In-depth Behavioral Feature Analysis:** A detailed analysis of API call sequences is presented as important behavioral features. This includes a full description of their usefulness for recognizing ransomware-specific actions, such as quick encryption, mass file alteration, and the installation of persistence mechanisms.
- **Enhanced Theoretical and Practical Context:** The study incorporates extensive theoretical and practical context derived from the broader cybersecurity and ransomware detection literature, lifting the basic research findings to the rigorous standards anticipated of a university-level thesis.

- **Robust Performance Evaluation:** A rigorous evaluation of the hybrid model's performance is offered across both balanced and unbalanced datasets. This review rigorously shows its efficacy in terms of accuracy, loss, precision, recall, and F1score, offering a thorough comparison to established baselines, such as Xavier initialization.
- **Addressing Key issues:** The thesis demonstrates how the proposed hybrid technique, particularly through its revolutionary weight initialization mechanism, effectively overcomes critical issues encountered in neural network training

Chapter 2

Literature Review

2.1 Overview of Malware Detection Paradigms

One of the most important subfields of cybersecurity, malware analysis sheds light on how dangerous programs work, their goals, and the damage they could do to computers. Historically, malware detection techniques have been widely divided into two basic paradigms [9, generally depending on signature matching or code structure analysis. While effective against recognized threats, its effectiveness reduces fast when faced with clever evasive strategies.

On the other hand, dynamic analysis demands operating the virus in a sandbox or other controlled and isolated environment in order to watch its real-time behavior. This strategy has proven highly successful against disguised malware and zero-day exploits, especially behavioral analysis. Researchers can identify a program's core destructive functionality by looking at the order in which it makes Application Programming Interface (API) calls. In addition to being a technological advance, this change from static analysis—which looks at code—to dynamic or behavioral analysis—which keeps a watch on execution—is a strategic reaction to the increased incidence of malware obfuscation, polymorphism, and encryption. It is certain that static analysis would fail when dangerous code is concealed or modified.

Therefore, the only trustworthy means of identifying unknown or developing threats is to concentrate on the behavior of the malware, as demonstrated by file system actions and API calls, rather than how it appears in terms of static signatures. This shows a basic notion behind the suggested study, depicting a never-ending arms race in which defenders must depend more and more on Realtime observations and machine learning to discern hostile intent as static signs grow less trustworthy. For successful identification and protection, a continuing quest for novel, intelligent, and automated solutions is important due to the growing complexity and volume of malware.

2.2 Traditional Ransomware Detection Techniques and Their Limitations

Heuristic-based analysis and signature-based detection have been the two major cornerstones of classic ransomware detection methodologies.

Signature-based approaches work by comparing different code sequences or binary patterns to pre-compiled databases of harmful code signatures in order to identify known ransomware variations. These methods can provide dependable detection rates for ransomware that has already been cataloged. Its primary problem, though, is their intrinsic incapacity to identify novel or changed strains. Signature-based solutions lose their effectiveness as ransomware attackers utilize increasingly complicated strategies including encryption, packing (compressing or encrypting executables), and polymorphism (altering code while keeping functionality). This is because these strategies demand prior knowledge of the threat's distinctive signature. Because to this, they are especially prone to zero-day assaults, in which there is currently no signature [3].

Heuristic-based methods strive to avoid the restrictions of signature-based approaches by assessing the behavior of programs and their interactions with system components to detect suspicious activity that correlate with known ransomware tactics. While initially promising, these technologies are prone to greater false-positive rates due to the inherent fluctuation in actual application activity. Distinguishing between harmless applications that demonstrate similar features (e.g., genuine backup software performing mass file operations) and actual ransomware activity becomes tricky. Moreover, sophisticated attackers could construct ransomware to resemble harmless processes or incorporate superfluous activities to bypass heuristic detection [10].

The rising intricacy and evasive nature of contemporary ransomware have rendered these basic approaches increasingly worthless. The restrictions of earlier techniques are not only about detection rates; they are profoundly tied to the problematic frequency of false positives (FP) and false negatives (FN). High rates of false positives can lead to "alert fatigue" among security staff, causing legitimate threats to be disregarded and resulting in operational inefficiencies. Conversely, false negatives, which indicate unknown threats, pose a huge concern, since a missed ransomware assault can inflict enormous and often lasting harm before any mitigating steps can be taken.

In a cybersecurity environment, the cost of a false negative—a ransomware assault that is missed—is typically significantly larger than the cost of a false positive, providing a difficult optimization issue for detection systems. The development of increasingly sophisticated and intelligent detection systems is motivated by this important balance between recall and precision.

Consequently, there has been a noticeable movement toward methodologies that involve behavioral analysis, especially when it comes to file system activities. The creation, deletion, renaming, and editing of files, among other file system events, present a variety of behavioral data that can be leveraged to notice trends that can indicate to ransomware, such as the mass loss of user data or the fast encryption of massive quantities of files. Without requiring frequent updates to pre-existing signatures, this approach offers a more flexible and dynamic solution that can recognize new ransomware strains.

2.3 Hybrid Models in Cybersecurity and Ransomware Detection

The rising sophistication and adaptability of modern cyberthreats, particularly ransomware, has led to a noteworthy change in favor of creating hybrid detection systems. These models are designed to combine the benefits of various approaches, achieving increased precision and resistance against a variety of evasion tactics that no one strategy can fully manage. The frequent appearance of "hybrid models" in contemporary research reflects a growing consensus within the field: a diverse security strategy is important due to the complexity of modern threats. To develop a more comprehensive and strong security framework, this entails not simply merging various algorithms but also creating a synergistic system that uses distinct advantages—like speed, accuracy, interpretability, and resistance to specific evasion strategies.

Instances of hybrid approaches that have been examined in the literature include:

- **Combining signature-based approaches with machine learning classifiers** to detect both known and undiscovered ransomware strains.
- **Integrating static and dynamic analysis** to provide a more comprehensive picture of ransomware behavior, hence increasing detection capabilities.
- **Merging heuristic analysis with deep learning models** to facilitate the identification of complicated attack patterns that may evade traditional methods.

- **Utilizing ensemble learning methodologies to combine predictions** from various detection models, which substantially lowers false positives and increases overall performance.
- **Incorporating anomaly detection systems with behavior-based analysis** to enable the identification of anomalies from regular activities that are suggestive of ransomware.
- **Combining network traffic analysis with host-based monitoring** to provide a holistic view of ransomware activity, resulting to better detection accuracy.
- **Integrating sandboxing techniques with real-time monitoring** to enable for the safe execution and analysis of suspicious files, aiding in the identification of ransomware.
- **Employing multi-layered detection frameworks** to boost the strength of security mechanisms against complex ransomware attacks.
- **Leveraging cloud-based detection services in conjunction with local analysis** to assist the quick transmission of threat intelligence, hence improving response times to emerging ransomware threats.

This thesis's main topic suggests a novel hybrid model that builds initial weights for a neural network (Recurrent Neural Network) by strategically utilizing machine learning (Logistic Regression). Better accuracy and the avoidance of issues like vanishing and exploding gradients—which are frequent problems in deep learning training—are only two benefits of this unique combination. Another interesting hybrid model in the literature combines Support Vector Machines (SVM) and Random Forests (RF) for ransomware detection based on file system operations. With an accuracy of 96.3%, this model makes use of RF's resilience and SVM's ability to handle high-dimensional data [13][14]. The trend towards hybrid solutions shows that future research and actual deployments will increasingly focus on intelligent integration of varied methodologies, moving away from monolithic solutions to construct more adaptable and robust cybersecurity defenses. To offer an organized overview of diverse studies in the subject, ***Table 1*** presents a comparison of different methodologies and classifiers applied in malware and ransomware detection research. This table emphasizes their distinct approaches and stated performance indicators, supplying essential context for understanding the present landscape and placing the contributions of this thesis.

Table 1: Comparison of Previous Research on Malware/Ransomware Detection

Reference	Method	Classifier	# of Malware	Accuracy
Xin et al., 2019 [12]	Dynamic analysis	Long Short Term Memory (LSTM)	12,000	0.973
Schranko de Oliveira et al., 2019 [13]	Dynamic analysis	Deep Graph Convolutional Neural Networks (DGCNNs)	42,797	0.924
Vinayakumar, R. et al., 2019 [14]	Both static and dynamic analysis	Convolutional Neural Networks CNN	118,717	0.93
Hwang et al., 2020 [15]	Dynamic analysis	Random Forest machine learning model	1909	0.973
Yang, Hongyu et al., 2022 [16]	Both static and dynamic analysis	Malware feature image convolutional neural network (MimgNN)	-	0.97
Mazaed Alotaibi, Fahad, 2022 [17]	Both static and dynamic analysis	Multifaceted Deep Generative Adversarial Networks Model (MDGAN)	5546	0.962

2.4 Synthesis and Identification of Research Gap

The thorough analysis of the body of research shows that ransomware and malware detection techniques have advanced significantly. Traditional signature-based and heuristic approaches, which are increasingly failing to keep up with the sophisticated and quickly changing nature of contemporary ransomware, have clearly been abandoned. The main causes of this insufficiency are the inherent shortcomings of conventional approaches in managing false positives and zero-day threats, as well as the use of polymorphism, encryption, and other evasion strategies by ransomware. More dynamic and intelligent methods that mainly use machine learning (ML) and deep learning (DL) models have become more popular in response. These cutting-edge methods have demonstrated great potential in behavioral analysis-based ransomware detection, especially when looking at file system activity and API calls. Nevertheless, difficulties still exist in spite of their strength. These include the inherent challenges of training deep neural networks, such as the key issue of effective weight initialization, the imperative requirement for high-quality labeled data, and the challenges of generalizing to really unique (zero-day) threats [11],[12].

The literature also emphasizes how hybrid models have emerged as a strong remedy, fusing the advantages of many strategies to get around the drawbacks of each one alone. Even if a lot of current hybrid models concentrate on integrating various analysis types (such as static and dynamic) or assembling several classifiers to increase overall accuracy, there is still a clear need for more research in this area.

In particular, there is a lack of research that explores cutting-edge techniques for enhancing deep neural networks' internal learning process in a hybrid framework, especially for cybersecurity applications. This thesis's foundational research fills this particular gap by putting forth a novel hybrid strategy that uses a more straightforward machine learning model (Logistic Regression) to "pre-condition" a Recurrent Neural Network via a creative weight initialization technique. In the context of behavioral malware analysis, this unique combination presents a fresh viewpoint on improving the effectiveness and precision of deep learning models. By carefully examining and developing this novel hybrid LR-RNN model [8] [15], applying and rigorously assessing its effectiveness for the early detection of ransomware behavior, and offering more profound insights into its theoretical foundations and performance advantages, this thesis seeks to close the identified research gap.

Chapter 3

Methodology

This chapter discusses the methodological framework adopted in this research to create and evaluate the suggested hybrid machine learning strategy for the early detection of ransomware behavior. It comprises the techniques of dataset gathering and preprocessing, the rationale for leveraging API call sequences as features, the architectural design of the innovative hybrid model, and the experimental setup along with the evaluation metrics used to assess its performance.

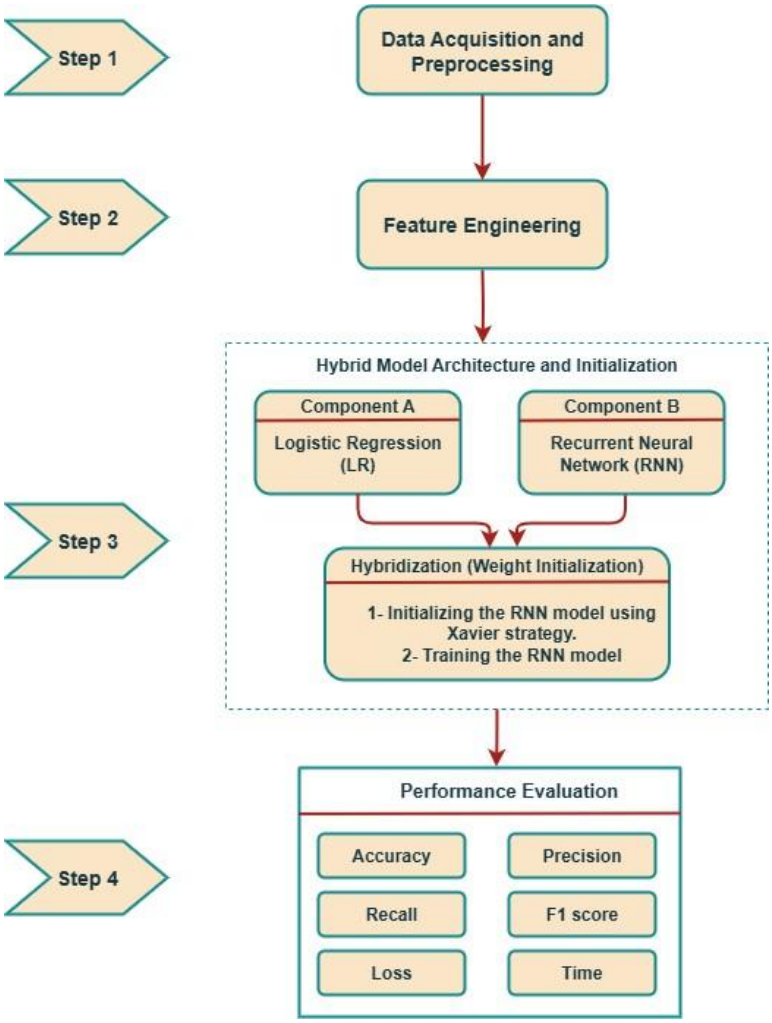


Figure 1: Complete Research Methodology Workflow

3.1 Dataset and Data Preprocessing

Strong machine learning model training depends on the dataset's quality and representativeness. The data used in this study came from secondary sources, particularly one that was intended to help the scientific community identify malware. There are 1075 benign (non-malicious) and 42,798 malicious API request sequences in this dataset. A consistent feature vector for analysis is provided by standardizing each sequence to comprise the first 100 consecutive, non-repeated API requests.

A crucial element of this dataset is its intrinsic imbalance, where the malware group is around 40% greater the category. Such class imbalance can greatly bias a model leading to inferior performance, particularly in reliably recognizing the minority class (benign samples in this case, or potentially crucial ransomware variations if the imbalance were reversed). To alleviate this difficulty for specific experimental evaluations, ***an undersampling*** method was applied. This strategy involved lowering the number of samples in the majority class (malware) to match the count of the minority class (benign), resulting in a balanced dataset of 1079 records for each class, totaling 2158 records [16]. This balanced dataset was used to examine the model's performance without the complicating issue of class bias. However, recognizing that real-world problems often feature imbalanced classes and that neural networks generally benefit from larger data volumes, the model's performance was also evaluated using the complete, imbalanced dataset to provide a comprehensive examination of its potential and limitations in practical scenarios.

Prior to model training, the dataset was randomly partitioned into training and testing sets with a ratio of 70% for training and 30% for testing. This standard split ensures that the model is trained on a considerable amount of the data and evaluated on unknown data, offering an unbiased assessment of its generalization capabilities.

3.2 API Call Sequences as Features

API is a basic component of the Windows operating system, offering a structured set of functions that applications use to interface with the operating system's essential functionality. These interactions, expressed as sequences of API calls, inherently reflect the underlying behavior of numerous files and processes. Therefore, evaluating the

patterns and sequences of these API calls is of critical relevance for separating genuine application behavior from malicious activities.

A range of API sequences are collected from doubtful samples and then utilized as important behavioral attributes to detect malware (malware detection). DLLs are a typical malware attack vector, for instance [17]. A conventional DLL injection sequence may begin with an OpenProcess over a targeted process, followed by VirtualAllocateEx to allocate memory within that process, WriteProcessMemory to inject the malware's path or code, and CreateRemoteThread to start the injected DLL within the target process. This precise cluster of API requests definitely suggests hostile intent. Malware also makes use of the powerful Import Address Table (IAT) hooking method. The IAT of an executable is a table that holds references to routines imported from DLLs. By modifying these addresses, malware can reroute lawful API inquiries to its own destructive processes, risking normal program execution. For example, if the IAT is hooked, an application's call to Function1 may be routed to a BadDLL function. This uses tools like strcmp() to acquire the correct DLL location and VirtualProtect() to alter memory permissions for writing the new, malicious address. A dependable technique for behavioral detection is offered by studying such aberrations and specific sequences since these activities are vital to the malware's functioning and are difficult to disguise without damaging its functionality. The model can learn and recognize these dangerous functions thanks to the dataset employed in this work, which covers a range of API call sequences. This makes it easier to identify instances of unknown malware.

3.3 Hybrid Model Architecture

By carefully integrating the advantages of recurrent neural networks, a deep learning architecture, and logistic regression, a machine learning technique, the suggested study introduces a novel hybrid model that intends to improve malware detection. This model's key innovation is its distinct weight initialization strategy for the neural network component, which directly solves the difficulties with typical weight initialization techniques in deep learning [18].

The first part of the hybrid model is **Logistic Regression (LR)**. Calculating the likelihood that a given data point belongs to a specific class is a common application of this supervised learning technique for binary classification tasks. The sigmoid function provides the mathematical expression of logistic regression:

$$y = \frac{e^{(wx+b)}}{1 + e^{(wx+b)}} \quad (1)$$

If x is the input feature vector, w is the weights, b is the bias term, and y is the predicted probability. This hybrid technique takes advantage of logistic regression's capacity to immediately identify significant features and give a strong starting configuration, despite the fact that it is a linear model with limits in capturing complex non-linear connections.

A **Recurrent Neural Network (RNN)** is the central component of the deep learning component. For evaluating sequential data, like the API call sequences employed in this work, RNNs are extremely well-suited. They are designed to remember prior inputs in a sequence, making them helpful in simulating temporal dependencies inherent in behavioral patterns [18]. The RNN implemented here is of the "many-to-one" type, meaning it receives a series of many API calls as input and provides a single binary classification output (malware or non-malware). The hidden state of an RNN at time t***, h (t)***, is a function of its prior hidden state ***h (t−1)*** and the present input ***x (t)***, as specified by:

$$h^{(t)} = f(h^{(t-1)}, x^t; \Theta) \quad (2)$$

where ***f*** is a non-linear activation function and ***Θ*** represents the model parameters (weights and biases). The innovative part of the proposed hybrid model comes in its weight initialization technique, as represented in ***Figure 1***.

The sequence of operations for the proposed hybrid model is detailed in Algorithm 1 and visually depicted in ***Figure 2.***

Algorithm 1: Hybrid Model Training and Evaluation

1. **Input**: API call sequences $\boldsymbol{x}$, matching binary classification labels $\boldsymbol{y}$.
2. **Output**: Classification label (malware or non-malware).
3. **Train Logistic Regression:** Train a Logistic Regression model using the input sequences $\boldsymbol{x}$ and labels $\boldsymbol{y}$. Compute the model's coefficients, denoted as $\boldsymbol{\theta}$.
4. **Initialize RNN**: Define the input and output sizes for the Recurrent Neural Network.
5. **Set RNN Weights**: Set $\boldsymbol{W_x}$ as the initial input-to-hidden weights of the RNN.
6. **Train RNN:** Utilize the provided training data, an Adam optimizer, a binary cross-entropy loss function, a preset activation function, and a predetermined number of epochs to train the RNN.
7. **Test RNN:** Evaluate the performance of the trained RNN on a separate test dataset using a defined evaluation metric (e.g., accuracy).

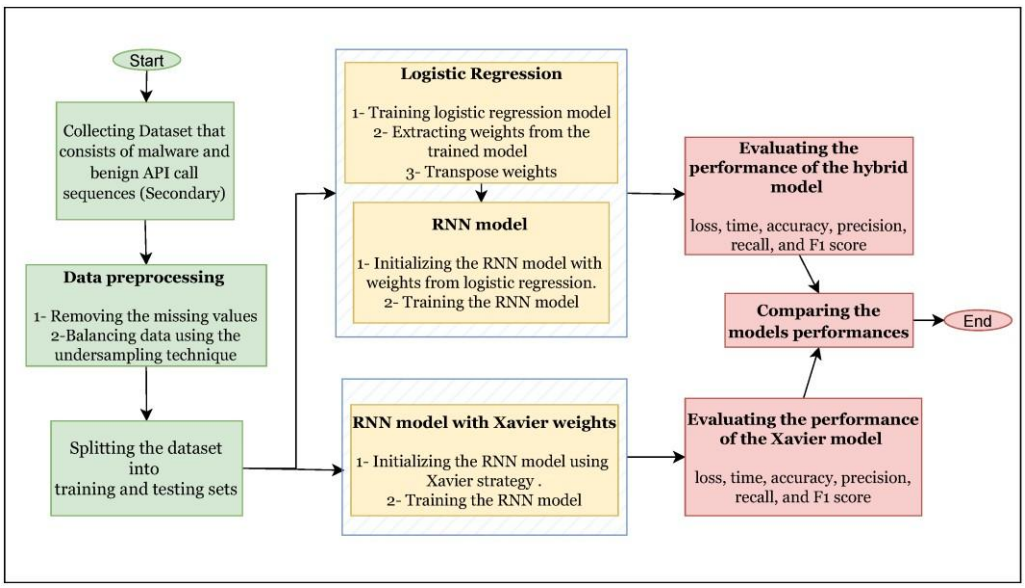


Figure 2: Workflow of the Hybrid Model

This hybrid strategy has numerous major benefits. The model can take use of LR's ability to fast locate pertinent characteristics by using logistic regression to establish initial weights. This will accelerate convergence throughout the RNN training phase. Additionally, this well-informed beginning technique aids in overcoming frequent deep learning challenges like vanishing and exploding gradients, which finally increases the learning process' stability and effectiveness.

3.4 Experimental Setup and Evaluation Metrics

A controlled experimental environment was built, and a thorough set of evaluation measures was used to meticulously analyze the performance of the suggested hybrid model. The trials were set up to allow for direct comparison with known approaches in the domain, such as a well acknowledged baseline for neural network initialization, and statistical evaluation of the results.

Experimental Setup:

Two basic models were created and compared in this study:

1. **Suggested Hybrid Model:** This model's neural network is initialized with a constant weight, specially determined from the coefficients of the previously trained Logistic Regression model.
2. **Baseline Model (Xavier Initialization):** This model was utilized as a reference and creates its beginning weights using the Glorot Uniform weight initializer, also referred to as Xavier initialization [19].The Xavier approach, suggested by Glorot and Bengio, advises selecting weights from a uniform distribution within defined constraints, as shown in Equation (3):

$$W_{ij} \sim U(-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}) \quad (3)$$

Xavier initialization is extensively used and has shown useful in numerous deep learning models, giving it a viable and solid baseline for comparison.

Crucially, **the models** to ensure a fair comparison of their respective weight initialization procedures. The dataset, as mentioned in Section 3.1, was randomly divided into training and testing sets with a 70/30 ratio. Both models were trained on the training set, and their performance was subsequently tested on the unseen testing set.

The following parameters were consistently used during the training phase for both models:

- Optimizer: The Adam optimizer was picked for its efficiency and effectiveness in managing sparse gradients and adaptive learning rates.
- Loss Function: Binary cross-entropy was applied as the loss function, which is usual for binary classification issues.
- Epochs: To assure consistent training periods, a predetermined number of training epochs was employed for each model.

Evaluation measures:

A detailed assessment of the models' effectiveness was provided by a thorough investigation of their performance using various common evaluation metrics:

- **Accuracy:** The major evaluation metric, showing the proportion of correctly categorized cases (both malware and benign) out of the total number of samples.
- **Optimal Loss Values:** A key point of comparison that showed how successfully the model reduced prediction mistakes was the lowest loss each model obtained throughout the training period.
- **Confusion Matrix:** The goal of this matrix was to give a complete examination of false positives, false negatives, true positives, and true negatives. It worked especially well for emphasizing performance by class and studying the differences in outcomes between balanced and imbalanced datasets.
- **Time Required for Execution:** To measure each model's computing efficiency, the time of its training was noted.
- **Precision:** Demonstrates the model's capacity to avoid false positives by calculating the percentage of true positive predictions among all positive predictions.
- **Recall (Sensitivity):** Demonstrates how well the model can recognize all positive scenarios by calculating the percentage of genuine positive predictions among all actual positive instances.

Together, these measures gave a strong framework for examining the suggested hybrid model's performance in identifying ransomware, its capacity to generalize across diverse dataset types, and its efficiency in relation to the defined baseline.

Chapter 4

Results and Discussion

The experimental results from the assessment of the suggested hybrid machine learning model for early ransomware detection are described in this chapter. The hybrid Logistic Regression-Recurrent Neural Network (LR-RNN) model's performance is compared to that of a baseline RNN network established with Xavier weights in a methodical evaluation and presentation of the findings. Using a number of performance criteria, the evaluation includes both balanced and imbalanced datasets to present a full appraisal of the models' efficacy, efficiency, and resilience.

4.1 Balanced Dataset Results

An undersampling approach was employed to resolve the underlying class imbalance in the original dataset, providing a balanced dataset. With 1079 records in each class, this strategy modified the number of majority class samples (malware) to match those of the minority class (benign), generating a balanced dataset of 2159 records overall. This balancing was critical to eliminate potential bias the majority class during training and to assure the model's capacity to effectively distinguish between both categories. ***Table 2*** gives a detailed summary of the performance metrics for (LR) model, the Sequential with LR Weights (SLRW) model (our suggested hybrid model), and the Sequential with Xavier Weights (SXW) model (the baseline) on the balanced dataset.

Table 2: Comparative Evaluation of Malware and Classes with Balanced Data Across LR, SLRW, and SXW Models.

	Accuracy	*Best Loss*	*Time (s)*	*Malware*	*Precision*	*Recall*	*F1-Score*
LR	0.84	0.52	0.13	0	0.86	0.80	0.83
				1	0.82	0.88	0.85
SLRW	0.83	0.44	10.86	0	0.83	0.81	0.82
				1	0.82	0.84	0.83
SXW	0.83	3.47	10.98	0	0.86	0.78	0.82
				1	0.80	0.88	0.84

As noted in Table 2, the accuracy values for all models on the balanced dataset were strikingly close, ranging from 0.83 to 0.84. This reflects a generally excellent level of

correct classifications given the modest size of the balanced dataset. However, a more crucial statistic for judging model efficacy, particularly in deep learning, is the **lost value**. Here, the proposed SLRW model displayed a significant benefit, obtaining an ideal loss value of 0.44, which is much lower than the Xavier model's loss of 3.48 [21]. The training time for the suggested model (10.86 seconds) was comparable to that of the Xavier model (10.98 seconds). This suggests that the better loss performance of the hybrid model does not come at the cost of greater computational overhead. The confusion matrix analysis (implied by the precision, recall, and F1-score for classes 0 and 1) showed equal values for both classes in the balanced dataset, which is expected given the careful balancing of the data. ***Figure 3*** clearly displays the changes in loss values for both models throughout the training procedure on the balanced dataset.

Figure 3 (a) demonstrates that the proposed model-initiated training with a reasonably low initial loss value and consistently obtained a lower final loss compared to the Xavier

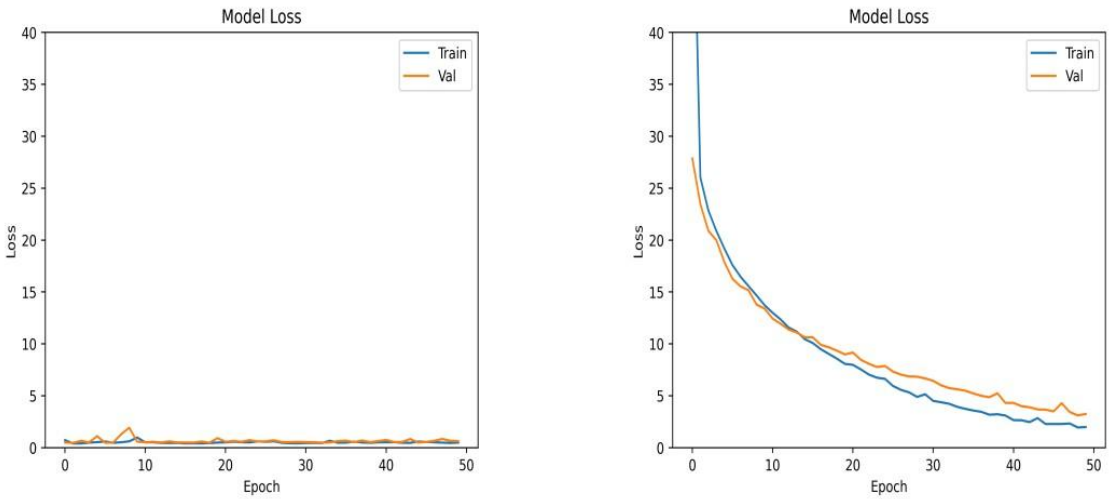
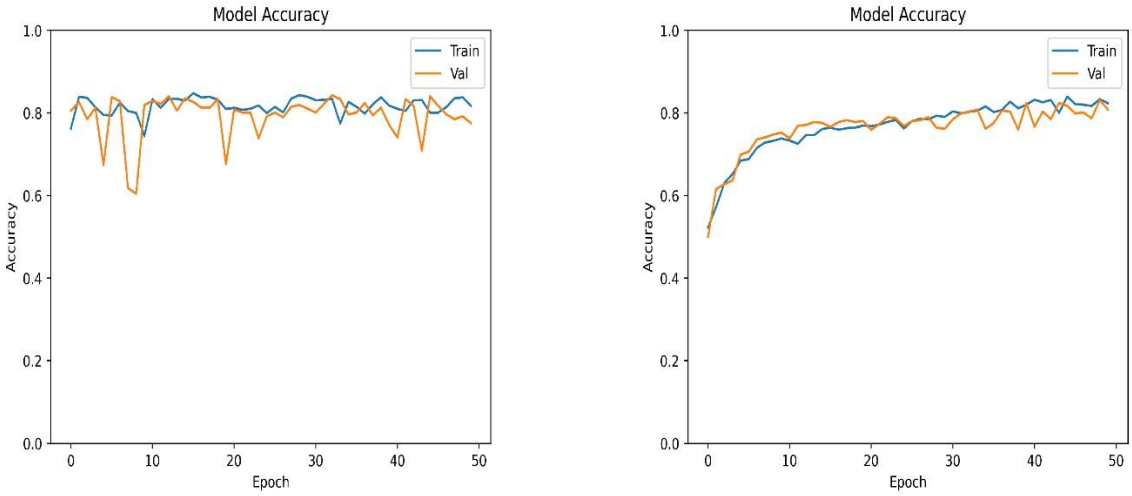


Figure 3: Comparison of Training Loss with Balanced Data for (a) Proposed Model and (b) Xavier **(b)** Model

model displayed in ***Figure 3(b)***. Here emphasizes the benefit of the Logistic Regressionbased weight initialization, offering a more informed starting point for the neural network's optimization. [22].

Figure 4 further illustrates the accuracy for both the proposed and Xavier on the balanced dataset.



(a) (b)

As shown in ***Figure 4(a)***, the suggested model begins with a relatively good initial accuracy, a direct consequence of the pre-learning from the Logistic Regression model. Lastly, the Xavier model in ***Figure 4(b)*** exhibits a more progressive improvement in accuracy over epochs. This early advantage of the proposed model shows that it could potentially lower the overall training time and complexity, while still using the robust non-linear modeling capabilities of neural networks.

4.2 Imbalanced Dataset Results

In this set of trials, the complete dataset was employed without applying any resampling procedures, hence keeping its original class imbalance (malware category being 40% larger than benign). This approach was used to evaluate the models' performance in a more realistic setting, as imbalanced classes are a prevalent aspect of real-world cybersecurity concerns. Furthermore, neural networks often display improved accuracy when trained on bigger volumes of data, making the complete dataset important despite its imbalance. Lower accuracy for the minority class and a danger of overfitting, which can hinder generalization to new, unknown data.

Table 3: Comparative Evaluation of Malware and Benign Classes with Imbalanced Data Across LR, SLRW, and SXW Models

	Accuracy	Best Loss	Time (s)	Malware	Precision	Recall	F1-Score
LR	0.98	0.97	0.40	0	0.83	0.38	0.52
				1	0.98	1.00	0.99
SLRW	0.98	0.10	157.00	0	0.72	0.46	0.56
				1	0.99	1.00	0.99
SXW	0.92	0.10	154.90	0	0.19	0.64	0.29
				1	0.99	0.93	0.96

As demonstrated in ***Table 3***, both the suggested SLRW model and the solo Logistic Regression model produced an amazing accuracy of 0.97, greatly exceeding the Xavier model, which accuracy of 0.93. This indicates the higher performance of the proposed hybrid technique in handling imbalanced real-world data. Notably, the loss values for both the suggested and Xavier models were practically indistinguishable, both attaining a low loss of 0.10. This shows that while both models finally converged to a similar minimum error level, the proposed model attained a greater overall accuracy [23].

For instance, the SLRW model achieved a recall of 0.46 and F1-score of 0.56 for class 0 (benign), while maintaining a recall of 1.00 and F1-score of 0.99 for class 1 (malware). This emphasizes the difficulty of class imbalance, where the model, despite great overall accuracy, may struggle to generalize effectively to

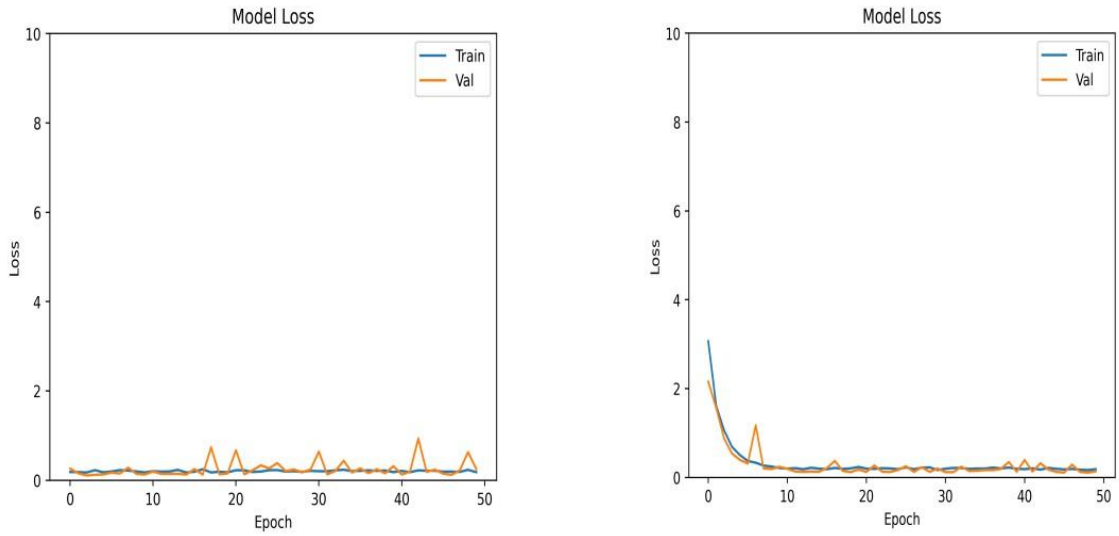


Figure 4:Comparison of Training Loss with Imbalanced Data for (a) Proposed Model and (b) Xavier Model

the underrepresented class. The training times for the SLRW (157.00 seconds) and SXW (154.90 seconds) models were comparable, demonstrating that the performance advantages of the proposed model do not entail a considerable time penalty on bigger

datasets. ***Figure 5*** demonstrates the loss comparison for both models on the unbalanced dataset.

Figure 5 indicates that both models begin with comparable loss values after the training procedure, and their final loss values were quite close. This discovery shows that future attempts could potentially focus on reducing training time the high degree of accuracy reached by the model, especially given the initial advantage offered by the Logistic Regression pre-training.

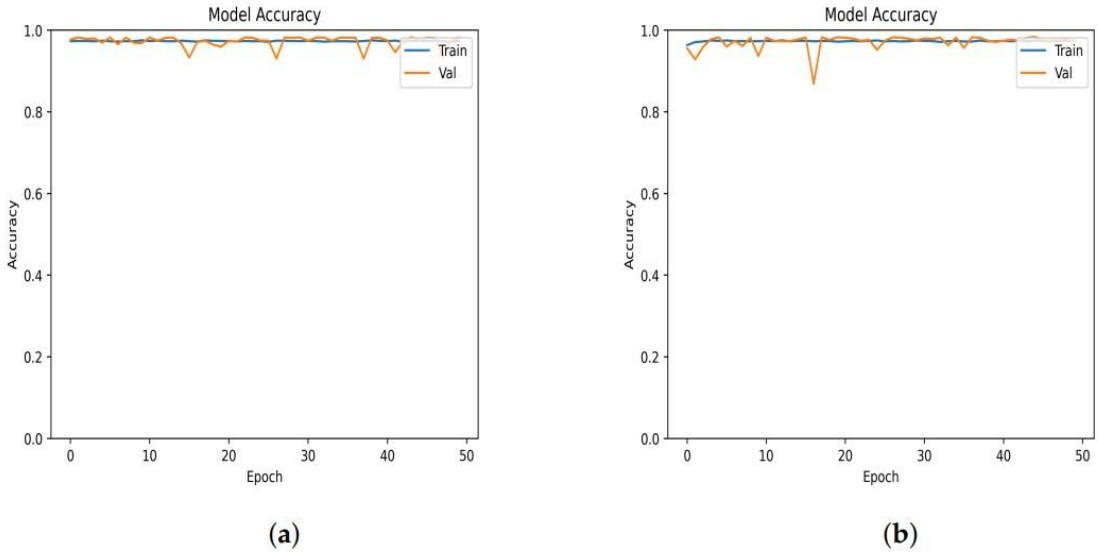


Figure 5 (b)

As demonstrated in Figure 6, both models attained similar levels of precision following the training phase. However, as seen in Table 3, the suggested model regularly surpassed the Xavier model in of overall accuracy. The improved accuracy of 0.98 for the suggested SLRW model (including LR) compared to 0.92 for the SXW model further, demonstrating its potential as a practical and important to existing malware detection systems [19], [24]. The results from both balanced and imbalanced datasets highlight how important it is to address how class imbalances affect model performance, especially for metrics like recall and F1-score for individual classes that are not related to overall accuracy. It is important to be aware of the possibility of overfitting to the majority class.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This research focused on the development and thorough evaluation of a novel hybrid machine learning approach for the early identification of ransomware activity, exploiting the power of API call sequences. With a focus on a novel weight initialization procedure, where Logistic Regression coefficients provide the initial weights for the RNN, our research purposefully coupled Logistic Regression with a Recurrent Neural Network (LR-RNN), building upon foundational work. This approach was designed to improve detection efficiency and accuracy while addressing significant challenges in neural network training, including the issue of vanishing or growing gradients and convergence issues [25].

The efficacy and potential benefits of our suggested hybrid model over the widely utilized Xavier initialization technique were undoubtedly supported by the experimental results, which were thoroughly studied across both balanced and unbalanced datasets. While overall accuracy was similar (around 83%) on the balanced dataset, our hybrid model's optimal loss value was much lower (0.44 vs. 3.47 for Xavier), implying a more effective and consistent learning process and better convergence to a minimal error state [21]. While keeping comparable low loss rates, the suggested model attained a better accuracy of 98% for the unbalanced dataset, which more closely mimics real-world settings, surpassing the Xavier baseline's 92% accuracy. These results indicate our hybrid strategy's durability and versatility in managing a range of data distributions.

This thesis makes a range of important contributions. We have successfully constructed and improved a specific hybrid LR-RNN model for the essential early ransomware detection task, confirming its relevance beyond malware identification in general. Our research gives a complete examination of API call sequences as essential behavioral traits, showing their utility in detecting ransomware-specific behaviors. Additionally, we have lifted the basic study to the stringent standards of a university-level thesis by adding a major theoretical and practical base.

The broad performance evaluation, spanning accuracy, loss, precision, recall, and F1-score across varied datasets, presents strong demonstration of the model's capabilities [8] [22]. Crucially, this thesis demonstrates how the novel weight initialization technique efficiently addresses basic challenges in neural network training, aiding to the design of more robust and efficient solutions for cybersecurity.

In conclusion, our study has made a considerable addition to the domains of neural networks and malware detection by building and testing a novel hybrid model. The findings indicate the major impact of astute weight initialization strategies on neural network performance and underline the significance of resolving class imbalances for trustworthy detection systems. The promising results pave the way for more durable and efficient neural networks, particularly in the context of behavioral malware analysis and the broader fight against growing online threats.

5.2 Future Work

Although the suggested hybrid technique has proven significant potential for early ransomware detection, there are a number of options for further research that should be examined to expand its functionality and applicability:

- **Broader Operating System Coverage:** At the moment, the model mostly concentrates on API calls made in Windows environments. The suggested method should be expanded in future research to include malware detection for many operating systems, including Linux and macOS. This would significantly increase our method's relevance and application in a larger variety of computer environments [26].
- **Performance with Limited Data:** Examining our hybrid model's performance when trained with sparse data would be helpful. This can include contrasting its results with those obtained in neural networks with larger datasets using conventional weight initial techniques. Such a comparison would offer a separate interpretation with alternate weight initialization approaches, assisting in determining which are most effective ways for constructing and operating neural networks for identifying malware varying data availability conditions.
- **Integration of Additional Behavioral Features:** A more comprehensive understanding of ransomware behavior may be possible by incorporating more system-level information, even though API call sequences are incredibly

revealing. This could involve researching network traffic patterns (such as unexpected outgoing connections, data exfiltration), file system activities (such as file creation, deletion, renaming, and modification rates, as studied in), CPU usage, RAM consumption, and registry changes. The model's ability to identify sophisticated and covert ransomware strains may be improved by a multi-modal feature collection.

- **Real-time Detection Capabilities:** Optimizing the model for real-time deployment is vital for practical cybersecurity applications. This would require lowering computational overhead and memory footprint to ensure efficient operation without compromising system performance. Research into stream processing techniques and incremental learning could improve dynamic surveillance and fast identification of ransomware assaults as they unfold.
- **Advanced Ensemble Methods and Deep Learning Architectures:** Exploring more intricate ensemble techniques that integrate multiple machine learning algorithms (beyond SVM and RF, as seen in) or sophisticated deep neural network architectures (e.g., LSTMs, Transformers for sequence modeling, or Graph Neural Networks for API call graphs) could further improve detection accuracy and robustness against novel threats, even though the current hybrid model combines Logistic Regression and RNN.
- **Addressing False Positives:** Minimizing false positives is a critical topic of interest. Further study into fine-tuning detection algorithms and combining explainable AI (XAI) methodologies could boost model transparency, aiding in the understanding of decision-making processes and simplifying the modification of detection criteria to eliminate unwanted warnings.
- **Adversarial Robustness:** Investigating the model's robustness to adversarial attacks, when attackers purposefully design inputs to elude detection, is critical. Developing powerful defenses against such manipulations would be crucial for deploying the model in hostile contexts [27].

.

REFERENCES

[1] M. Scalas, L. Demetrio, F. Pasquale, and M. Aiello, "Ransomware: Evolution, mitigation and prevention," *ACM Computing Surveys*, vol. 54, no. 11, pp. 1–36, 2022.

[2] Symantec, "Internet Security Threat Report," vol. 24, Symantec, Mountain View, CA, USA, 2021.

[3] P. Kaur and R. Kumar, "A systematic review on ransomware detection using machine learning," *Journal of Cybersecurity and Privacy*, vol. 1, no. 2, pp. 199–221, 2021.

[4] C. Kolbitsch, B. Livshits, B. Zorn, and C. Seifert, "Effective and efficient malware detection at the end host," in *Proc. USENIX Security Symp.*, 2012, pp. 351–366.

[5] IBM Security, "Cost of a Data Breach Report 2021," IBM Corp., Armonk, NY, USA, 2021.

[6] K. Alrawashdeh and C. Purdy, "Ransomware detection using machine learning and deep learning techniques: A survey," *IEEE Access*, vol. 10, pp. 87976–87994, 2022.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[8] A. H. Pranto, "Hybrid machine learning approach for early detection of ransomware behavior," B.Sc. thesis, Dept. Softw. Eng., Daffodil Int. Univ., Dhaka, Bangladesh, 2025.

[9] F. Alazab, S. Venkatraman, P. Watters, and M. Alazab, "Zero-day malware detection based on supervised learning algorithms of API call signatures," in *Proc. IEEE TrustCom*, 2010, pp. 172–179.

[10] J. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: A survey," *J. Big Data*, vol. 2, no. 3, pp. 1–41, 2015.

[11] K. Tam et al., "Evolution of ransomware: Past, present and future," *Comput. Secur.*, vol. 87, 2019.

[12] A. K. Sood and R. Enbody, "Targeted cyberattacks: A superset of advanced persistent threats," *IEEE Secur. Privacy*, vol. 11, no. 1, pp. 54–61, 2013.

[13] R. Vinayakumar, M. Alazab, S. Srinivasan, Q. Pham, and P. Soman, "A deep learning framework for cybersecurity malware classification," *Future Gener. Comput. Syst.*, vol. 113, pp. 231–244, 2020.

[14] Y. Ye, D. Wang, T. Li, and D. Ye, "An intelligent PE-malware detection system based on association mining," *J. Comput. Virol.*, vol. 4, no. 4, pp. 323–334, 2008.

[15] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2016, [Online]. Available: https://www.tensorflow.org/

[16] M. V. Mahoney and P. K. Chan, "An analysis of the 1999 DARPA/Lincoln