SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I9P118

Volume 12 Issue 9, 203-218, September 2025
© 2025 Seventh Sense Research Group®

Original Article

Intelligent Ransomware Detection Using Hybrid CNN-Bidirectional GRU with Optimized Training and Low Computational Overhead

M.S. Balamurugan[1*], V. Rajendran[2], S. Suma Christal Mary[3]

[1,2]*Department of Electronics and Communication Engineering, Vels Institute of Science Technology and Advanced Studies, Chennai, Tamil Nadu, India.*
[3]*Department of Artificial Intelligence and Data Science, Chennai Institute of Technology, Chennai, Tamil Nadu, India.*

[1]*Corresponding Author : balams1982@gmail.com*

Received: 18 July 2025 Revised: 20 August 2025 Accepted: 19 September 2025 Published: 29 September 2025

Abstract *- The significant increase in ransomware assaults, which peaked over the past decade till 2024, makes it extremely concerning for cyber experts to track early detection methods continuously. This ransomware virus remains one of the most significant threats governments and businesses must confront. Conventional signature-based anti-ransomware solutions and heuristic-based and rule-based methods often struggle to identify ransomware malware, which is ineffective at detecting known threats. Various researchers used machine learning techniques for ransomware detection, leading to a lack of reliability in real-world scenarios and higher computational time costs. To tackle the challenge of ransomware detection, this research work focuses on a deep learning-based hybrid model that combines CNN-LSTM, CNN-GRU, and CNN-Bidirectional GRU. Each layer effectively trains the parameters to detect malware. The CNN-Bidirectional GRU model achieved a maximum accuracy of 99.8% when using the Adam and RMSProp optimizers, with a computational cost of only 0.01 seconds. Using these optimizers, the proposed model reached a greater convergence rate, which protects the files against Ransomware. Additionally, comparisons were made between traditional machine learning and deep learning methods across various metrics, including training accuracy, validation accuracy, training losses and validation losses, to evaluate the overall performance of the proposed methods.*

Keywords *- Bidirectional Gated Recurrent Unit (BID-GRU), Convolutional Neural Network (CNN), Cybersecurity, Gated Recurrent Unit (GRU), Ransomware malware.*

1. Introduction

Several sub-families of cyberattacks have previously been identified, including crypto, locker, and scareware. In recent times, various malware such as Doxware, leakware, and Ransomware as a Service (RaaS) have affected cybersecurity [1]. Crypto-ransomware encodes files and data on a system without affecting normal computer operations, rendering the material unreadable and lacking a decryption key. The effects of this assault are extremely difficult to retrieve when data encryption methods included in crypto-ransomware are used appropriately [2]. The main goal of locker malware is to prevent computers from operating properly. Victims are typically presented with a ransom request via an authentication screen that includes a countdown timer to expedite their activation. In contrast to crypto-ransomware, Locker ransomware attacks are typically preventable by using a virus scanner and safe mode restarts [3]. Scare-ware is a type of Ransomware that pretends to be a virus or other computer issue and uses false advertising to trick people into downloading harmful software, even though it does not actually damage the victim's machine [4]. Often, this Ransomware floods the screen with pop-up notifications, frightening users into downloading software that demands payment to restore the issues [5].

According to the Analysis conducted by CISCO [6], cybercrime activities peaked in January and continued to rise for the remainder of the observation period, as depicted in Figure 1. The patterns observed in the dropper area and Ransomware strongly resembled one another, indicating a relationship between the two. Droppers were probably being utilized to spread the ransomware payload. Because it generates money by locking data and systems captive for ransom, Ransomware is still a common danger. Because of its great profitability and growing number of ransomware-as-a-service platforms, even less experienced attackers may already start attacks.

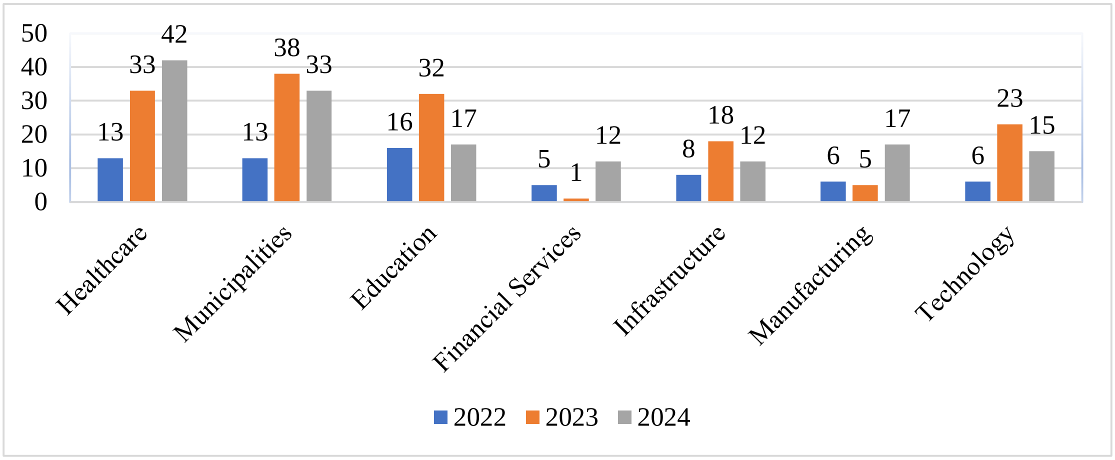

Fig. 1 Cisco ransomware statistical analysis during 2022-2024

The cycle of attacks is sustained by organizations' frequently insufficient safeguards and recovery procedures, as well as the readiness of numerous parties to give up the blackmail [7]. In April 2024, 373 more instances of ransomware malware were reported.

Most of the current research concentrates on characterizing ransomware kinds and their technological methods. Specifically, there has been less focus on detecting new variations early on, when conventional signature-based methods cannot offer sufficient security.
The research gap is to establish an explainable AI model, researching ransomware behavioral patterns, or developing a machine learning–based detection system. Also, developing a thorough strategy that increases resistance, improves early detection, and lessens the overall impact of ransomware assaults on vital systems by methodically examining various attack sub-families.

The main drawbacks of conventional techniques, especially signature-based methods, include altered ransomware versions, which are distinct from existing malware "fingerprints," such as file hashes, byte patterns, or known payloads, and are impossible to identify. Furthermore, rule-based approaches rely on preset rules that need to be manually developed and adjusted by professionals; these rules cannot change on their own. Heuristic-based approaches, which are imprecise for changing threats, leverage statistical characteristics or behavioural assumptions to indicate various suspicious activities, such as instantaneous file transformation or unexpected API access. Such malware causes financial losses to organizations, institutions, and software companies. It is crucial to use early detection technology, such as a proposed hybrid deep learning-based optimization technique, to recover with the least amount of danger while remotely keeping backups at secure locations to stop [8] the massive financial losses that firms suffer as a result of ransomware attacks. The main contribution of this research is as follows;

- The primary goal of this study is to identify ransomware malware using the CNN-LSTM, CNN-GRU, and CNN-Bidirectional GRU approaches in the context of developing technologies.
- Proposed hybrid deep learning optimization methods successfully identified ransomware malware. This research work utilized datasets that had been collected from open sources, of which 80% was used for training and 20% for the testing phase to evaluate the CNN-LSTM model that had been constructed.
- Optimization algorithms, such as SGD, RMSProp, and Gradient Descent, provide optimal solutions with a greater convergence rate for ransomware detection, which enhances the effectiveness of the deep learning model.
- Without requiring intricate feature engineering, the proposed hybrid deep learning model attained a 99.8% accuracy rate with a loss rate of 0.01, reducing computational cost. This leads to the detection of ransomware malware, which is appropriate for avoiding cyberspace.

By disseminating dangerous software and stealing personal data, cybercriminals threaten organizations, governments, and individuals globally [8]. Hundreds of hackers use harmful software throughout the day to break into networks, steal information, and conduct illegal financial activities.

As a result, the scientific community is becoming increasingly concerned about the protection of personal data. To detect dangerous software and stop it from accessing private data, this study uses deep CNN techniques and hybrid (CNN-LSTM, CNN-GRU, CNN-BiGRU) models [9]. To provide a reliable and efficient method for ransomware malware classification and detection, the authors compared machine learning, hybrid deep learning models, and optimization methods to overcome the challenges faced in signature-based, rule-based, and heuristic-based techniques.

2. Related Work

Several methods have been used to identify and categorize ransomware malware. The literature contains a large number of studies on automatic malware detection, Analysis, and categorization [10, 11] and dynamic Analysis performed on malware [12] as described in Table 1. However, machine learning and deep learning development have created many new opportunities for computer malware identification and categorization, as done in various research [13]. Aljabri et al. [14] detected ransomware malware, including Lockbit, Revil, and BlackCat, utilizing 48 features, including memory features, with an accuracy rate of approximately 98%. Several authors detected ransomware malware via Android application Alsoghyer et al. [15], classification based on n-grams of opcodes [16], malware behaviors identified [17], machine learning techniques applied by Bae et al. [18], Noorbehbahani et al. [19] and Poudyal et al. Applied static level analysis on ransomware features [20] also used k-fold cross validation for better malware prediction, Gbenga et al. [21] used eight machine learning algorithms, among which optimization based Extra Tree and Random Forest chose random features for predicting malware, Smurf attack- a kind of DDoS malware detected by Revathy et al [22]. Bibi et al. [23] performed multifactor feature iterations along with a Recurrent Neural Network, an optimization approach carried out with ANN by Abdolrasol et al. [24], malware behaviors analyzed using a deep Neural Network by Tobiyama et al. [25], and image-based malware detected using the CNN-transfer learning model by Pant and Bista [26]. Deep learning-based feature extraction was unlabelled along with a Pareto ensemble classifier for malware classification by Zahoora et al. [27]. An Entire Analysis of these review papers uses heuristic techniques to find irregularities in system activity or concentrates on static or signature-based Analysis to find known ransomware families. Even though these methods have proven effective against conventional Ransomware, they frequently miss new variations, polymorphic strains, or assaults Delivered as-a-Service (RaaS), which change quickly to avoid detection. Furthermore, a lot of earlier research focused on classification accuracy without delving deeper into the characteristics or actions of the system that point to ransomware activity. This research implemented a deep learning based hybrid CNN with LSTM, GRU, and Bi-GRU models with parameter tuning, providing greater accuracy, less time duration, and the least computational overhead.

Table 1. State-of-the-art techniques on ransomware detection

Survey	Year	Dataset	Methodology applied	Work Intention	Limitations	Accuracy	FPR	Precision	Execution time	TPR/Detection rate	ROC AUC
[14]	2024	Ransomware (lockbit, revil, blackcat)	XGBoost	48 memory features are extracted	Leads to maximum Execution time	97.85%	0.022	97%	NA	NA	96.9 %
[18]	2020	Signature-based malware	Decision tree model	Develop a protection mechanism	Focused on only the specific operation of Ransomware	98.5%	0.03	98%	NA	NA	NA
[20]	2018	Ransomware-based malware	Machine learning techniques	CF-NCF values	Predicted the malware falsely	97%	0.2	96%	NA	96%	NA
[21]	2021	Malware prediction	Random Forest + Extra Tree	Only dynamic features were analyzed	Maximum computational overhead	98.5%	1.4	NA	NA	98.09%	NA
[22]	2022	Smurf attack	Hybrid (DT+RF)	Detect DoS malware in an efficient manner	Lower detection rate	98%	NA	97.3 %	0.01	NA	NA
[25]	2016	Ransomware	Deep Neural Network	Predict the behaviors of malware while using a website or URL	Maximum computational time period	96%	NA	NA	NA	NA	92%
[27]	2021	Zero-day ransomware-based malware	Pareto ensemble-based estimator approach	Enhance accuracy and recall. Solve class	Focused on host-based features	93%	NA	93.4 %	NA	93%	NA

				imbalance							
[32]	2017	Ransomware data samples	LSTM	Dynamic malware analysis	Applied pretrained models, a large quantity of API calls, hence a delay in execution	96.6%	NA	NA	NA	NA	NA
[33]	2023	Ransomware	CNN+Pre-trained transformer	The ensemble model improves model performance	Computational complexity more	98.9%	99%	NA	NA	98%	NA

The literature survey analysis suggests that machine learning models are ineffective at detecting Ransomware, as multiple authors failed to assess parameters, including detection rate, TPR, and ROC AUC. In addition, traditional approaches, primarily signature-based, rule-based, and heuristic-based models, were unable to identify suspicious activities like file access and modification of API calls. These are the challenges faced by various researchers in detecting ransomware malware. Hence, this research employed deep learning models, optimization techniques, and hyperparameter tuning, providing better optimal solutions with less computational cost, faster convergence, maximum accuracy, and minimum loss.

3. Methodology

3.1. Data Collection

The ransomware dataset has been gathered from the open resource website GitHub [28], whose link is mentioned below, where the contributors obtained examples of ransomware samples. In this instance, the metadata comprised 138,047 input samples, which were further separated into valid (41,323) samples classed as Class 1 and ransomware (96,724) records classified as Class 0. This indicates that there is an imbalance in the data samples, with more ransomware samples than benign samples.

3.2. Data Balancing

This section provides an overview of how ransomware samples are distributed via class distribution, applying the resampling method by which imbalanced data are balanced and fed into the next process. Table 2 reveals that the data is unbalanced, causing the employed model to overpredict Ransomware, and the precision will be reduced. This leads to bias if the model is always trained to predict the majority classes.

In this case, the data samples are balanced by undersampling Ransomware and oversampling the benign samples with the SMOTE technique. The representation of class distribution before balancing samples and after balancing samples is shown in Figure 2 and Table 2.

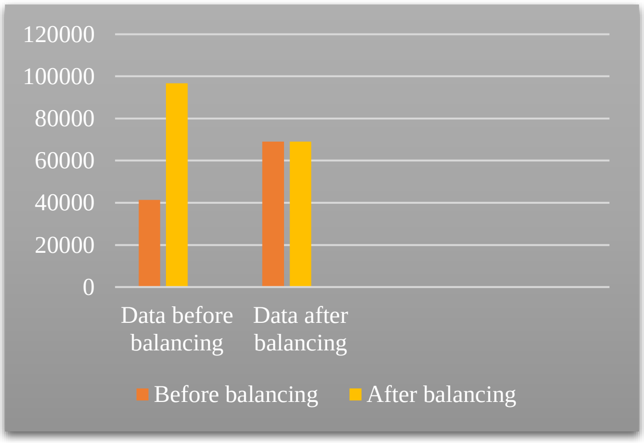

Fig. 2 Class distribution before and after balancing

Table 2. Class distribution on ransomware dataset

Class	Count
Ransomware	96,724 samples
Benign	41,323 samples

3.2.1. Data Collection

The input samples were distributed into two classes: class 0 comprised 1521 samples, and class 1 included two samples. Moreover, resampling techniques were applied to mitigate class imbalance in ransomware data samples. Deep-learning workflows require the use of resampling algorithms to handle a variety of issues with data quantity, quality, and dispersion. These methods include methodically adjusting the training dataset to enhance standardization and overall model performance. The reason for using resampling techniques in this research is to address the class imbalance among malware samples and enhance model generalization via data augmentation [29]. To address these issues of class imbalance, oversampling minority classes and undersampling the majority classes were done[30]. This work carried out an Oversampling approach for minority classes: the Synthetic Minority Oversampling technique (SMOTE), suitable for boosting the number of occurrences in minority classes by creating fake samples or replicating preexisting samples.

Perform the sampling method for the majority classes: To balance the distribution of classes, the number of ransomware samples is reduced, which is considered as under-sampling of the majority classes. In some cases, there is a chance that significant data will be lost when using random undersampling methods. Therefore, the authors might not have utilized the random under-sampling approach.

3.2.2. Sample Ransomware Dataset Statistics

Initially, the malware dataset is imbalanced, with unequal distribution of target values among distinct labels. The statistics of the ransomware dataset are listed in Table 3, and the resampled metadata are listed in Table 4.

Table 3. Initial dataset statistics

	1001	**1**	**2**	**0**	**0.1**
Count	1523.0	1523.0	1523.0	1523.0	1523.0
Mean	16806.21	0.3814	2.029	0.296	0.003
Std	4882.53	0.4859	3.166	0.456	0.057
Min	10002.0	0.00	0.0	0.000	0.00
25%	10807.0	0.00	0.00	0.00	0.00
50%	20232.0	0.00	0.00	0.00	0.00
75%	20754.0	1.00	3.00	1.0	0.00
Max	21259.0	1.00	11.00	1.0	1.00

Table 4. Resampled Dataset statistics

	1001	**1**	**2**	**0**	**0.1**
Count	3042.0	3042.0	3042.0	3042.0	3042.0
Mean	18679.83	0.19	1.016	0.399	0.001
Std	3944.26	0.39	2.459	0.489	0.04
Min	10002.0	0.00	0.0	0.00	0.00
25%	20102.0	0.00	0.0	0.00	0.00
50%	20228.5	0.00	0.0	0.00	0.00
75%	21019.0	0.00	0.0	1.00	0.00
Max	21259.0	1.00	1.00	1.00	1.00

3.2.3. Covariance Matrix for Sample Data

The covariance values between two samples in a random vector are described by a specific type of matrix called the covariance matrix. The coefficient of variation appears between the non-diagonal values, and the variance of every value represented along the primary diagonal of the matrix is considered as a variance-covariance matrix. The covariance matrices for the ransomware samples are listed in Table 5.

Table 5. Covariance matrix

	10001	**0**	**0.1**
10001	1.0634	-0.0756	0.1127
0	-0.0756	0.974	-0.022
0.1	0.1127	-0.0221	0.9621

The covariance between the two variables, X1 and X2, can be evaluated using Equation (1).

$$Covariance\ (P,Q) = \frac{1}{n}\sum_{i=1}^{n}(P_i - \bar{P})(Q_i - \bar{Q}) \qquad (1)$$

Where n indicates the number of data samples, $P_i\, P_i$ and $Q_i\, Q_i$ represents individual sample points, $\bar{P}\bar{P}$ and $\bar{Q}\bar{Q}$ denote the mean values of P and Q. Moreover, the covariance matrix of ransomware metadata with n values is a $n\ x\ n$ $n\ x\ n$ matrix, where every data element $Cov_{i,j}\, Cov_{i,j}$ indicates the covariance between two variables, i and j, as denoted in Equation (2).

$$Covariance\ matrix = \begin{bmatrix} Var(P_1) & Co(P_1,P_2) \ldots & Cov(P_1, Pn) \\ Cov(P2, P1) & Var(P2) \ldots & Cov(P2, Pn) \\ Cov(Pn, P_1) & Cov(Pn, P_2) & var(Pn) \end{bmatrix} \qquad (2)$$

3.2.4. Analysis of Correlated Attributes

A heatmap is a graphic display of data in which colors correspond to different values. Heatmaps are employed throughout the deep learning pipeline for ransomware detection to aid in comprehending feature correlations, data trends, model performance, and system behavior. The representation of the heatmap is depicted in Figure 3 to analyze the correlated attributes in the given dataset.

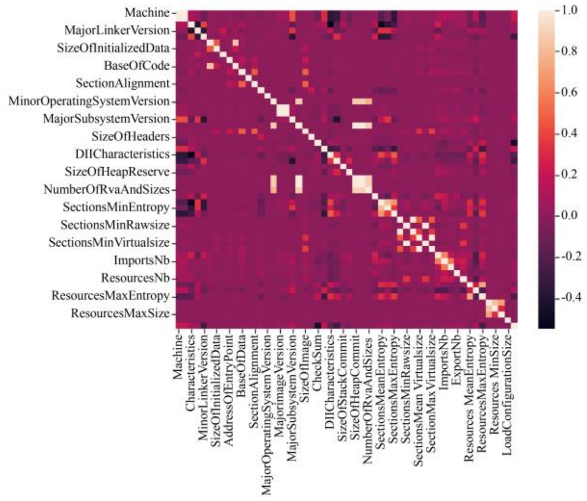

Fig. 3 Class distribution before and after balancing

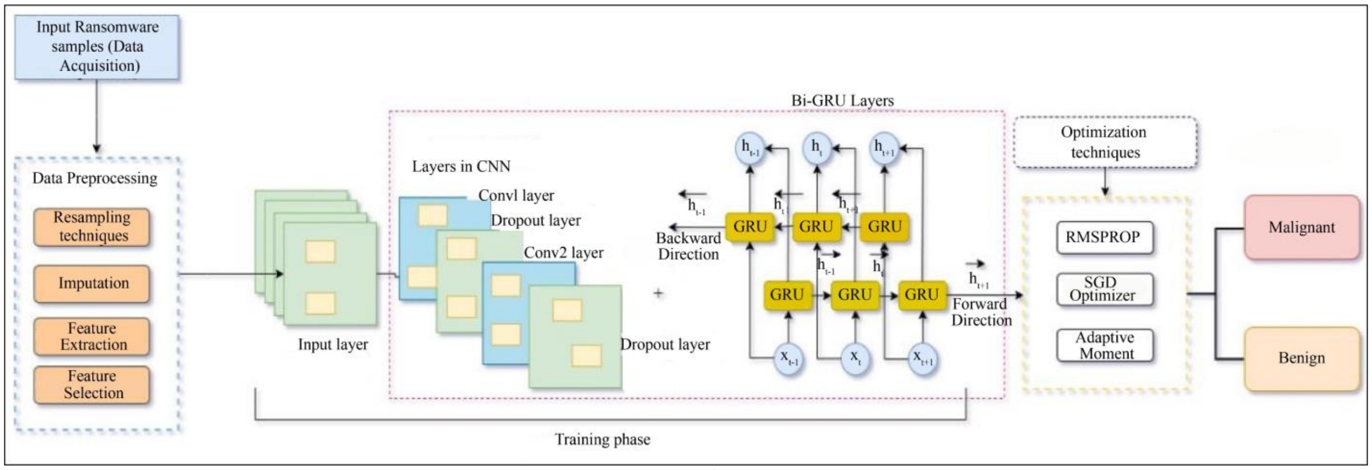

Fig. 4 Proposed framework

3.3. Data Preprocessing

An essential first step in developing efficient ransomware detection systems is the preprocessing phase. Deep-based models effectively extract and learn relevant features from raw data and must be transformed clearly and organized [31]. The adaptability, performance, and accuracy of malware detection models were enhanced by appropriate preprocessing. The cornerstone for developing powerful ransomware detection methods involves efficient data preprocessing. This proposed detection models can perform and be more reliable if you carefully gather, clean, transform, and prefer pertinent data to solve issues such as confidentiality and inconsistency. These preparatory measures are integrated with suitable deep-based CNN methodologies and continual examinations that enhance their security against ransomware attacks.

3.3.1. Data Standardization

The process of converting input samples into a standardized appearance that has a mean value of 0 and a standard deviation value of 1 is known as data standardization. This evolution keeps features with different sizes from overwhelming those with greater dimensions, ensuring that every feature contributes equally to the model's learning process. Standardization can be described scientifically as mentioned in Equation (3).

$$z = \frac{x-\mu}{\sigma} \tag{3}$$

x indicates the input sample, μ specifies the mean value of the features, σ is the standard deviation, and z is the standardization form. By ensuring that each feature contributes evenly, the standardization method enhances the convergence rates and overall performance. Standardization strengthens accessibility and precision, and supports model training in ransomware detection, where data can be high-dimensional and heterogeneous. Through comprehension and application of suitable standardization methods, utilization of suitable instruments, and adherence to optimal protocols, the efficacy of the ransomware detection method is considerably enhanced.

3.3.2. Reordering Method

A crucial aspect of data preprocessing for malware detection is reordering. It places data in a different order, structure, or arrangement to improve the efficiency and performance of machine learning models that are used to detect and categorize harmful software. Rearranging data correctly can increase the generalization to new data, decrease computing complexity, and improve model accuracy.

3.3.3. Imputation Technique

The technique of substituting values for incomplete or inadequate elements in a dataset is known as data imputation. This procedure is vital for preserving the accuracy of the statistical Analysis because imprecise data can result in imbalanced findings and reduce the reliability of the dataset. The above-mentioned techniques used acceptable values obtained from patterns found in the existing ransomware data to replace missing items. This made it possible to conduct a deeper and more precise evaluation. Because several machine learning methods are unable to deal with such missing numbers directly, resolving these values is crucial to prevent biased findings or failure to execute the algorithm. A single estimated worth is used in a single imputation to substitute all missing data in the ransomware dataset. Compared with multiple imputation methods, this method is faster to use. They disregard the ambiguity surrounding the imputation procedure and instead handle the imputed values as true values. Several techniques, such as regression-based and hot-deck imputations, are applicable to resolve the missing values; however, in this study, mean, median, and mode imputation approaches were primarily used. Mode, Median, and Mean Imputation: This process involves using the mean, median, or mode of the dataset's accessible data points to fill in the missing values. These techniques are simple to use; however, they run the risk of adding biases and warping the original distribution of the dataset.

3.4. Splitting of Data

Dataset splitting is a crucial stage in the deep learning workflow to ensure that the models are trained, verified, and tested efficiently to attain outstanding efficiency and adaptability. The authors created reliable deep CNN hybrid models that function well in ransomware detection by comprehending and applying appropriate splitting strategies, following best practices, and avoiding typical mistakes. Following feature selection, the samples were divided between the training and testing phases using an 80:20 ratio. In this case, 20 data samples were verified to assess the deep learning model performance in ransomware detection and categorization from legitimate, while the remaining 80% of data samples were trained to forecast Ransomware that impacts devices and files. Additionally, the impact of the technique was measured using training and testing times.

3.5. Hybrid CNN with Bi-GRU Framework for Ransomware Prediction

Maniath et al. [32] utilized a deep-based Long Short-Term Memory approach for Ransomware to classify API calls into binary classes. Singh et al. [33] employed a RANSOMNET+ tool comprising a CNN with pre-trained transformers for malware classification. Zhang et al. [34] introduced a patch-based CNN for ransomware family detection, based on n-grams of opcodes. In this study, a hybrid-based CNN and a Bidirectional GRU were employed to predict and classify malware in an optimal manner. Convolution layers are the hidden layers observed in CNNs. Convolution layers, which drive across the input weights and change the neuron input on the activation function, are the fundamental idea of CNN construction. The hybrid framework approach is illustrated in Figure 4. A pooling layer, many conv2d layers, and a dropout layer can occasionally comprise a convolution layer. This results in a large number of conv2d layers being crowded into two layers. A classification layer processes the output after it has been flattened into a single vector of length.

The model's accuracy and score were estimated after splitting the training into 80% and testing to 20%. The outcome of the experiment is a prediction of whether a file is malicious.

3.5.1. CNN-LSTM Framework

In addition to relationships between variables, such as reliance, uniformity, and structural stability information in the ransomware data gathered, the hybrid CNN-LSTM method eliminates the requirement for laborious feature engineering, which is necessary for shallow machine learning techniques, but may not be able to extract sufficient useful features for the task of classification [35]. When trained on the same input, CNNs can detect particular characteristics at higher layers and select representations from various perspectives. While Recurrent Neural Networks (RNNs) are particularly effective at collecting sequential data, Convolutional Neural Networks (CNNs) are excellent at detecting local spatial correlations. The layers used for the hybrid CNN-LSTM model in this study are listed in Table 6.

Table 6. Layers in CNN-LSTM

Layers	Filters	Pool size	Kernel size	Function
CNN layer				
Conv1D	64	-	3	ReLU
MaxPooling1D	-	2	-	-
Conv1D	128		3	ReLU
MaxPooling1D	-	2	-	-
LSTM layer				
LSTM	100	-	-	-
Dropout	0.5	-	-	-
LSTM	50	-	-	-
Dropout	0.5	-	-	-
Dense	1	-	-	Sigmoid

3.5.2. CNN-GRU Framework

Gate mechanisms that are particularly well-suited to handling time-sequential tasks define the GRU neural network. Such Gate devices are simplified in recurrent cells to enhance the computational efficiency in an attempt to conserve the ransomware prediction performance of the LSTM network. As shown in Figure 5, the GRU artificial neural network comprises two control gates, known as the reset and update gates (Z_{t}). The amount of data that must be erased from the hidden state of the preceding instant of time is determined by the first gate, also known as the Reset Gate (Rt). Whenever the number was near zero, the facts of the prior moment were discarded. If the number is close to one, concealed data associated with an earlier moment is maintained inside the present storage.

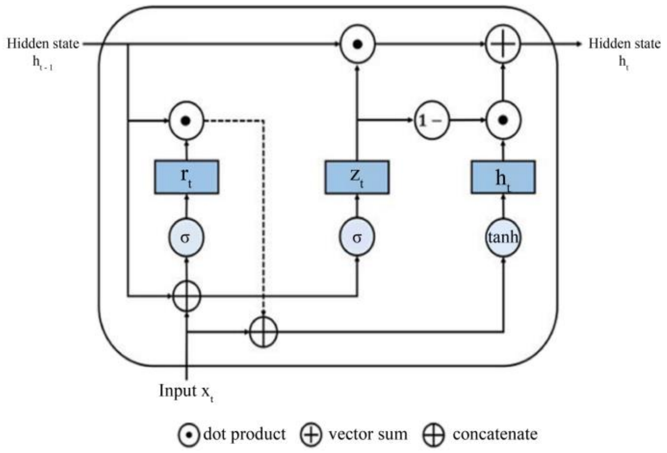

Fig. 5 Framework for GRU

The update gate (Zt), which is also known as the subsequent gate, transfers the quantity of data from the prior instant's concealed state to the present concealed state. In this situation, if the numerical value is close to 0, the data from the preceding instant's concealed state are disregarded; however, if the number is close to 1, the data are maintained in the present concealed state (Ht-1). The training time of this model is greater than that of the LSTM and Bidirectional GRU, which leads to greater limitations in the GRU model.

$$Z_t = \sigma\,(W_z * [h_{t-1}, x_t]) \qquad (3)$$

Σ represents the sigmoid function available in the activation layer, $\tilde{H}t$ indicates the hidden state, and the Activation function, along with the fully connected layer, represented as σ.

3.5.3. CNN-Bidirectional GRU Framework

The proposed hybrid CNN and a Bidirectional GRU is a deep neural network that uses a sequence processing approach. It consists of two GRUs: one GRU receives input ransomware samples in the forward direction and the other receives them in the backward direction. This model is suitable for classifying ransomware malware and for determining malignant data between the source and destination sides in both directions—layers in the CNN-bidirectional GRU. The layers utilized in the CNN model and the Bidirectional GRU are listed in Table 7.

Table 7. Layers in CNN-Bidirectional GRU

Layers	Filters	Pool size	Kernel size	Function
CNN layer				
Conv1D	64	3	ReLU	-
MaxPooling1D	-	-	-	2
Conv1D	128	3	ReLU	-
MaxPooling1D				2
GRU layer				
GRU	100	-	-	-
Dropout	0.5	-	-	-
GRU	50	-	-	-
Dropout	0.5	-	-	-
Dense	1	-	Sigmoid	-

The Adam optimizer and sigmoid were used as activation functions because this is a binary sequence classification issue. The highest training accuracy of 99.8% was achieved with batch sizes of 64 and a learning rate of 0.01 and 10 epochs.

3.6. Optimization Techniques

The main purpose of optimization techniques is distinct, even though they provide a means of minimizing the loss function based on the training dataset using deep learning models. Moreover, optimization focuses on identifying an appropriate model given a limited quantity of data to reduce the training error. To detect and classify ransomware malware efficiently, the authors employed optimization approaches such as Adam, SGD, and RMSProp by tuning hyperparameters in terms of learning rate, batch size and epochs. There are several challenges in deep learning optimization techniques, such as local and global minima, through which errors can be minimized.

A local minimum may exist for any objective function f(x) if the value of f(x) at x is less than the values of f(x)at any other points in the vicinity of x.f(x)is the global minimum if its value at x is the objective function's minimum over the whole domain. The local and global minima can be represented as Equation (4).

$$\mathrm{f(x)} = \mathrm{x} \cdot \cos\cos(\pi \mathrm{x}) \text{ for } -1.0 \le \mathrm{x} \le 2.0 \qquad (4)$$

3.6.1. Adam Optimizer

As predicted, this method is gaining prominence as one of the more reliable and successful optimization techniques for deep learning. Here, the model is trained with 10 epochs, or 10 iterations through the training data, using the Adam optimizer with a learning rate of 0.01. To provide classification probabilities for every malware data point, a sequential model was selected to run on the complete dataset. Each convolutional layer comprises filter sizes ranging from 64 to 128. The final samples were placed in two successive thick layers using three convolutions. These layers work similarly to a multilayer perceptron, producing an output vector based on the activation function by multiplying the weights of malware samples by the kernel function. A specific percentage of the network neurons was dropped using drop layers to avoid overfitting the training data. The probability of the input samples falling into each of the two classes was calculated by the last output layer, SoftMax, and the dense layer produced the output in the form of two classes: Class 0(malignant) and Class 1 (normal).

3.6.2. Stochastic Gradient Descent (SGD) Optimizer

SGD optimizer suitable to determine the average of the loss functions for each case in the training dataset. By mentioning that, given a ransomware training dataset of 'n' samples, the loss function is denoted as fi(y) with respect to learning sample indexing 'i and y indicates the features. The objective function of SGD can be estimated using Equation (5).

$$\mathrm{f(y)} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i=0}}^{\mathrm{n}} \mathrm{f_i\,(y)} \qquad (5)$$

The gradient (∇) of the above function can be calculated as mentioned in Equation (6).

$$\mathrm{Af(y)} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i=0}}^{\mathrm{n}} \mathrm{Af_i\,(y)} \qquad (6)$$

If gradient descent is employed, the computational cost for each iteration of an independent variable is O(n), and it increases linearly with n. As a result, gradient descent will cost more per iteration when the training dataset is larger.

Every iteration of SGD lowers the computational expenses. Ransomware samples I ∈ {1,…,n} were chosen randomly to represent the examples at each stage of stochastic gradient descent, and then calculate the descent gradient ∇ fi(y) for updating x as in Equation (7).

$$y \leftarrow y - \eta \nabla f_i(y) \qquad (7)$$

The learning rate is denoted by η. By observing that the gradient descent's O(n) computational cost decreases to the constant O(1) with each iteration. Full gradient estimation can be represented as Equation (8).

$$E_i \nabla f(y) = \frac{1}{n} \sum_{i=0}^{n} \nabla f_i(y) = \nabla f(y) \qquad (8)$$

The SGD optimization algorithm trained the deep learning models faster in analyzing system behavior, changes in features, and API calls. Through analyzing such abnormal behaviors, the models were trained to update model weights using small batches of data, thereby reducing the computational cost for every iteration while predicting ransomware malware with optimal solutions.

3.6.3. RMSProp (Root Mean Square Propagation)

A potent optimization technique in machine learning (RMSprop) determines the system features that best correlate with the actual values as well as model predictions. An adaptive learning rate optimization approach called RMSprop was created to speed up the gradient descent performance. This analogy illustrates the optimization of a loss function to determine the ideal model parameters for the best match between the actual and predicted values. To avoid leakage of average weighted samples, this momentum method is suitable for obtaining the optimal solution in ransomware malware detection, in which a few samples are greater than zero, fixing other samples remain static. 'Consequently, using St (samples in the vector state) with a reasonable distribution of Gt (gradient squares) enhances the convergence rate in the detection and classification of ransomware malware, as in Equations (9) and (10).

$$S_t \leftarrow \gamma S_{t-1} + (1-\gamma) G_t^2 \qquad (9)$$

$$X_t \leftarrow X_{t-1} - \frac{\eta}{\sqrt{S_t + \epsilon}} \odot G_t \qquad (10)$$

3.7. Metrics Evaluation

There are four distinct quadrants in the generated confusion matrix: a) True Negative (Quarter on the Top Left), b) top-right quadrant of false positives, c) False Negative (quarter on the bottom-left), and d) (Bottom-Right Quadrant) True Positive. False indicates a mistake or incorrect prediction, whereas true indicates that the values were correctly anticipated. After creating a confusion matrix, various metrics were evaluated to assess the quality of the model. The confusion matrix for the CNN-BiGRU model using the Adam optimizer, where True Positive=187, False Positive=96, True Negative=13, and False Negative=9, is illustrated in Figure 6.

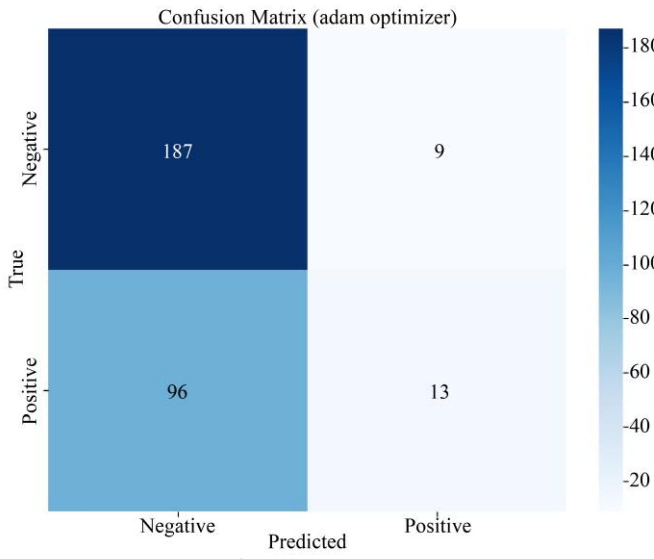

Fig. 6 Confusion matrix (adam optimizer)

Several metrics, namely precision, recall, F-score, and accuracy measures, were used to evaluate the overall performance of the deep-based CNN-LSTM, CNN-GRU, and CB-GRU during both the training and testing stages. The ransomware samples that were perfectly categorized fit the actual class as Truly Positive (TP), the number of samples wrongly distinguished fit the actual value as False Positive FP, the predicted value of data perfectly distinguished, which are unfit to the class Truly Negative (TN), and the output value of ransomware samples wrongly classified unfit to actual values as False Negative FN.

Precision indicates the ratio of truly identified positive samples from the predicted positive samples described in Equation (11).

$$\text{Precision} = \frac{\text{Truly predicted as positive}}{\text{True positive} + \text{False Positive}} \qquad (11)$$

Recall: calculated as the ratio of truly predicted samples as positive to the combination of true positives and false negatives using Equation (12).

$$\text{Recall} = \frac{\text{Truely predicted as positive}}{\text{True Positive} + \text{False Negative}} \qquad (12)$$

Accuracy: Accuracy represents the correct prediction of ransomware malware for all input samples. It is defined as the total number of correctly predicted samples divided by the total number of samples, as described in Equation (13).

$$\text{Accuracy} = \frac{\text{No. of samples predicted truly as Ransomware}}{\text{Overall input data samples}} \qquad (13)$$

The loss-negative average among ransomware samples was evaluated using Equation (14). To predict the entire loss while training the parameters, the binary cross-entropy function is calculated, as mentioned in Equation (15).

$$\text{Log loss} = \frac{1}{N}\sum_{i=1}^{N}(\log(P_i)) \qquad (14)$$

$$\text{Log loss} = \frac{1}{N}\sum_{i=1}^{N} -(y_i \text{ x } (\log(P_i) + (1 - y_i \text{x}\log(1 - P_i)) \qquad (15)$$

Here, binary classification of ransomware malware was found, in which P_i indicates the probability of class 1, whereas (1-P_i) represents the probability of class 0. True Positive Rate (TPR)- TPR is also called the detection rate, a key metric to measure the effectiveness of ransomware malware detection. TPR can be calculated as Equation (16).

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (16)$$

TP indicates True Positive, where malware is correctly identified as malware, whereas FN indicates False Negative, where malware is incorrectly identified as a normal sample.

3.8. Hyperparameter Tuning

Improper tuning provided the least accuracy, and the losses in the features were higher. Therefore, the authors performed this study. For each metadata to be tuned to estimate the probability accurately, a different set of parameters is required. Such tuning consists of two steps: (a) fine-tuning hyperparameters, including learning rate, activation function, number of nodes in the network, preferred optimizer, batch size, and number of epochs, and (b) neural network layer tuning. Using a variety of optimization approaches, including Adam, RMSProp, and Stochastic Gradient Descent optimizers, which are listed in Table 8, the hyperparameters, including the input size, batch size, filter size, kernel size, number of epochs, learning rate and variation in dropout, were adjusted for model development.

Table 8. Hyperparameter tuning via optimizers

Models	Input size	Batch size	(Filters, Kernel size)	Learning rate	Epochs	Optimizers
CNN-LSTM	(100, 64)	32	(64,3) (128,3)	0.01	1-10	SGD, Adam, RMSProp
CNN-GRU	(50, 128)	64	(64,3) (128,3)	0.02	1-10	SGD, Adam, RMSProp
CNN-Bidirectional GRU	(100, 64)	128	(64,3) (128,3)	0.03	1-10	SGD, Adam, RMSProp

4. Experimental Outcomes

When compared to the SGD optimizer, RMSProp reduces the manual tuning of parameters, such as the learning rate and batch size. For hybrid models such as CNN-LSTM, CNN-GRU, and Bidirectional GRU, the input size has 100-time steps with 64 features, and then the size is reduced to (50, 128), meaning 50-time steps with 128 features in Tables 9 and 10. From the table below, Adam optimization approach reaches validation accuracy of 59.84, loss 0.6 ms and True Positive Rate 60%.

As per Table 10, various metrics are evaluated in which training loss reaches 0.7 for both SGD and RMSProp optimizers, whereas accuracy reaches approximately 60%, with a lower convergence rate and a lesser detection rate of 11%; hence, the SGD and RMSProp optimizers did not achieve better results.

Table 9. Detection of ransomware using CNN-LSTM with adam optimizer

Epoch	Training loss	Training accuracy	Validation loss	Validation accuracy	TPR
1	0.67	60.44	0.66	59.84	58
2	0.66	61.35	0.67	59.84	52
3	0.66	61.35	0.67	59.84	57
4	0.66	61.35	0.67	59.84	52
5	0.66	61.35	0.67	59.84	54
6	0.66	61.35	0.67	59.84	60
7	0.66	61.2	0.66	59.84	58
8	0.66	61.08	0.66	59.84	54
9	0.65	62.35	0.66	59.84	54
10	0.65	61.9	0.66	59.02	55

Table 10. Comparison of CNN-LSTM using SGD

Epoch	Training loss	Training accuracy	Validation loss	Validation accuracy	Detection rate
1	0.686	59.16	0.68	59.02	11
2	0.68	60.8	0.7	59.02	11
3	0.7	60.8	0.7	59.8	11
4	0.7	61.02	0.7	59.8	11
5	0.7	61.08	0.7	59.84	11
6	0.7	61.2	0.7	59.8	11
7	0.7	61.1	0.7	59.8	11
8	0.7	61.2	0.7	59.8	11
9	0.7	61.2	0.7	59.8	11
10	0.7	61.2	0.7	59.8	11

Figure 7 illustrates the evaluation of ransomware prediction using various metrics.

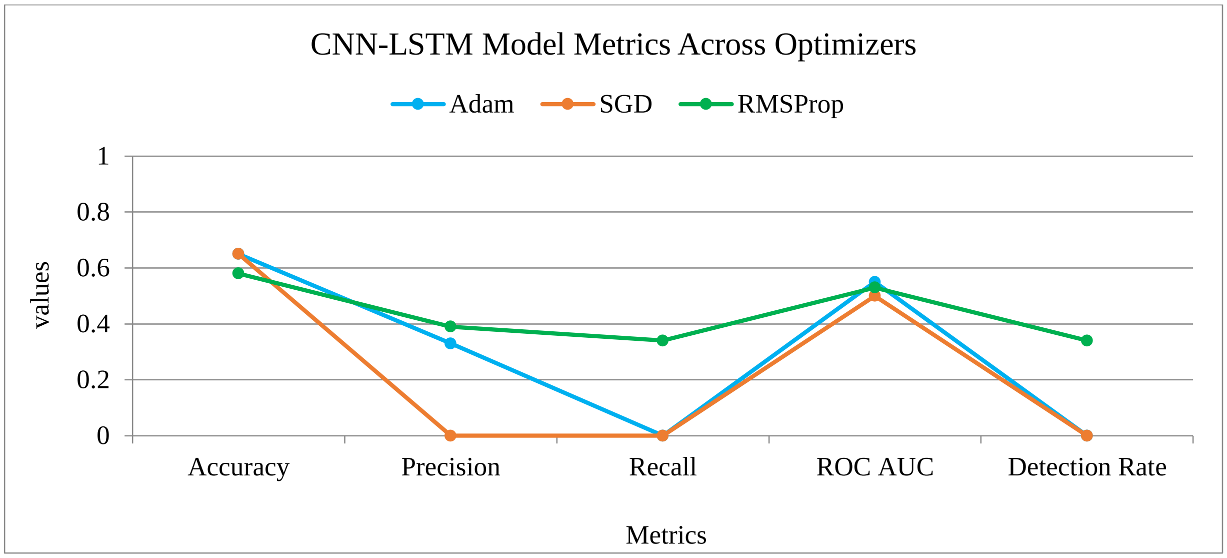

Fig. 7 CNN-LSTM model across optimizers

In terms of metrics, such as accuracy, precision, recall, ROC Metrics evaluation, such as ROC-AUC, and detection rate, using the hybrid CNN-LSTM model with Adam, SGD, and RMSProp optimizers.

Table 11. Comparison of CNN-LSTM using RMSProp optimizer

Epoch	Training loss	Training accuracy	Validation loss	Validation accuracy	Detection rate
1	0.67	59.8	0.7	59.84	11
2	0.7	61.4	0.7	59.8	11
3	0.7	61.4	0.7	59.8	11
4	0.7	61.4	0.7	59.8	11
5	0.7	61.4	0.7	59.8	11
6	0.7	61.4	0.7	59.8	11
7	0.7	61.35	0.7	59.8	11
8	0.7	61.35	0.7	59.8	11
9	0.7	61.35	0.7	59.8	11
10	0.7	61.35	0.7	59.8	11

4.1. Layers in CNN-Bi-Directional GRU Model using Optimizers

Table 12 and Figure 8 present the training and validation performance metrics of the CNN–Bi-Directional GRU model over 10 epochs using optimized hyperparameters.

Table 12. Layers in the CNN-Bi-directional GRU model using optimizers

Epoch	Training loss	Training accuracy	Validation loss	Validation accuracy	Detection rate (s)
1	0.6	68.9	0.6	71.3	99.8
2	0.5	78.5	0.3	85.3	99.5
3	0.2	93.9	0.06	98.36	99.7
4	0.07	97.7	0.02	99.18	99.8
5	0.01	99.5	0.02	99.18	99.8
6	0.005	99.8	0.02	100	99.8
7	0.002	99.8	0.02	99.9	99.8
8	0.002	99.9	0.02	99.9	99.6
9	0.001	99.9	0.001	99.8	99.8
10	0.001	99.9	0.002	99.8	99.8

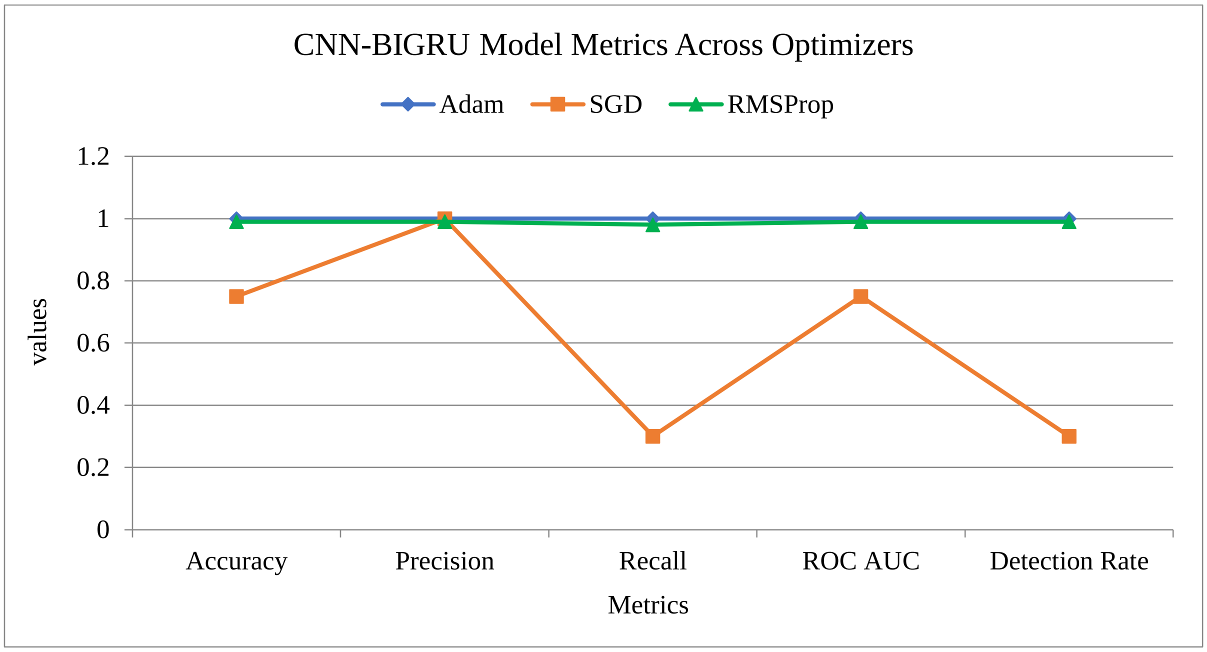

Fig. 8 Metrics evaluation using CNN-BiGRU model

4.2. Experimental Analysis and Statistical Results

Using the Jupyter notebook integrated development platform and the Intel Core I5 1.7GHz processor speed, the research was conducted mostly with the TensorFlow 2.3.1 Keras library. The authors analyzed the ransomware malware dataset by performing preprocessing, in which data balancing and imputation were performed, splitting the dataset into training and testing, building machine learning and deep-based hybrid models such as CNN-LSTM, CNN-GRU, and CNN-Bidirectional GRU. Validation was carried out to detect and classify the ransomware malware into benign and malignant. Among this hybrid model, CNN-Bidirectional GRU with Adam and RMSProp optimization techniques achieved a maximum accuracy, precision, recall, and detection rate of 100% with the least losses of 0.01, which provides a greater solution for predicting malicious attacks in a network environment. The evaluation of machine learning classifiers with imbalanced data and after resampling the data in terms of accuracy, logarithmic loss, and ROC-AUC score is described in Table 13.

Table 13. Assessment of machine learning classifiers

Model	Accuracy	Logarithmic Loss	ROC-AUC score
IMBALANCED DATA			
Logistic Regression	88.4	0.62	0.94
Multilayer perceptron	90.35	0.20	0.94
Random Forest	92.65	0.24	0.92
RESAMPLED DATA			
Logistic Regression	85.5	0.72	0.93
Multilayer perceptron	87.6	0.31	0.94
Random Forest	89.9	0.38	0.94

However, a machine-based random forest model achieved a greater accuracy of 92.65%, and after resampling, it was deduced to be 89.9% with 0.38 losses. By focusing on the maximum efficiency and least computational time in malware detection, the authors employed a deep-based hybrid model with optimization approaches to obtain an optimal malware detection and classification solution. The statistical Analysis conducted during this work, using various hybrid models along with three optimization models, is presented in Table 14. Predicting the detection rate enables the improvement of response time, evaluation of performance, and fine-tuning of parameters.

Table 14. Evaluation of a deep-based optimization approach

Models	Accuracy	Precision	Recall	ROC-AUC	Detection rate
CNN-LSTM	63.93	0.333	0.0092	55.97	92%
CNN-GRU	64.2	63.20	63.00	46.91	11.3
CNN-Bidirectional GRU	64.26	64.32	63.9	55.3	11.8
After optimization (CNN-Bi Directional GRU better outcomes)					
CNN-LSTM (Adam)	100%	99.9%	99.9%	100%	98%
CNN-LSTM (SGD)	64.26	0%	0%	49.32%	95%
CNN-LSTM (RMSProp)	57.7%	39.36%	33.9%	53.03%	33.94%
CNN-GRU (Adam)	65.57%	59.09%	11.93%	55.97%	11.93%
CNN-GRU (SGD)	64.26%	0%	0%	46.91%	65%
CNN-GRU (RMSProp)	64.26%	0%	0%	55.3%	82.5%
CNN- Bidirectional GRU (Adam)	99.5%	99.6%	99.6%	99.5%	99.8%
CNN-Bidirectional GRU (SGD)	74.43%	100%	28.4%	74.7%	28.4%
CNN-Bidirectional RMSProp	99.6%	99.4%	99.5%	99.8%	99.4%

Variants of Ransomware continue to develop; hence, their actions may differ significantly according to the type of attack, such as distinct encryption techniques or attack methods. The CNN-LSTM and CNN-GRU models cannot predict ransomware malware, in which the parameters are trained on particular malware traits, which makes it challenging to generalize to novel ransomware strains owing to multiplicity. This research employed the Adam optimizer, RMSProp optimization with CNN-bidirectional GRU, and sigmoid as the activation function to resolve the binary sequence classification issue. With batch sizes of 32, 64 and 10 epochs, the highest training accuracy of 99.4% was achieved. A comparison of the hybrid CNN-BiGRU and CNN-LSTM models across various optimizers is shown in Figure 9.

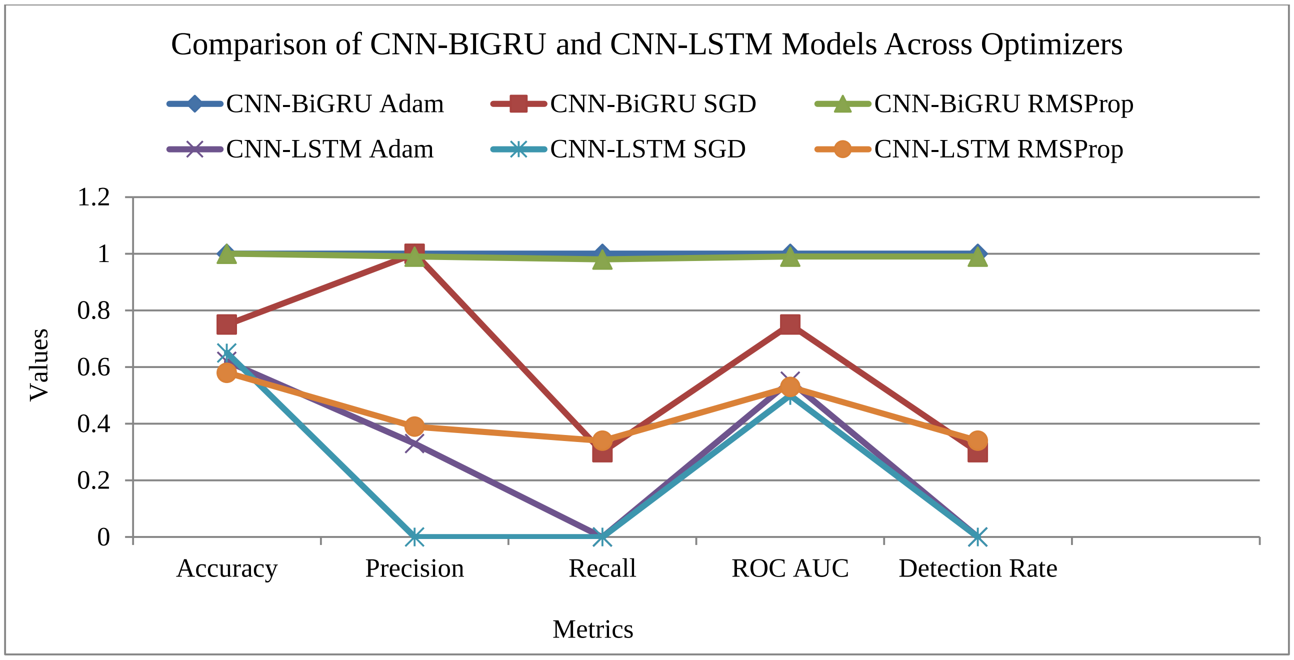

Fig. 9 Comparison of CNN-BiGRU and CNN-LSTM models

4.3. Comparison of the Proposed Framework with Traditional Approaches

Table 15 shows the assessment of various traditional approaches, such as machine learning and deep learning, in ransomware detection in terms of various metrics, such as validation accuracy, precision, recall, losses, and execution time. Various investigators achieved a maximum accuracy of 99%, while others attained 95% and 97%, respectively, but the solutions were not optimized, resulting in a lower convergence rate for the ransomware malware prediction. Hence, this research work proposes a hybrid optimization approach that effectively provides optimal solutions for detecting ransomware malware earlier, with fewer losses and a maximum accuracy of 99.8%, at a low computational cost.

Table 15. Analysis of ransomware detection done by various investigators

Investigator	**Samples**	**Approaches**	**Metrics Measure**				
			Accuracy	**Precision**	**F1-score**	**Loss**	**Execution Time**
Maniath et al. [32]	Ransomware	LSTM	96.67%	-	-	-	2 hours
Dipendra et al. [26]	Image-based malware	Custom CNN Transfer learning	98.7%	98%	99%	0.07-0.01	NA
Singh et al. [33]	Ransomware attack	Transfer learning	99.1%	99.5%	98.5%	0.03	NA
Malak et al. [14]	Ranso ware samples (Lockbit, Revil, Blackcat)	XGBoost	97.85%	97.1%	97.5%	-	NA
Zahoora et al. [27]	Ransomware malware detection	Ensemble model	93%	93%	94%	0.8	18 sec
Proposed CNN	Detects suspicious activities	Hybrid (CNN+BiGRU) using Adam Optimizer	99.5%	99.4%	99.5%	0.01	10 ms

5. Conclusion

The emphasis on Ransomware among numerous hackers has accelerated its progression and led to complex replications that can avoid detection by signature-based antivirus software. Consequently, it is crucial to deal effectively with both new families and new variations in recognized families. This study demonstrates that a hybrid deep CNN with a Bi-GRU model effectively detects ransomware malware and its families. In conclusion, CNN and Bi-GRU together achieved 99.8% accuracy with 0.01 loss rate in ransomware malware prediction under static conditions, and CNN-bidirectional GRU outperformed CNN-GRU and CNN-LSTM. Several literature surveys state that model performance was greatly impacted by incorrect hyperparameter tuning. To improve such performance in ransomware malware detection, hyperparameters such as batch size, learning rate, and epochs are adjusted in various optimizers like Adam, SGD, and RMSProp. Through these hybrid deep learning optimization models, cybersecurity experts now have a reliable method to combat ransomware threats. In future, interpretability frameworks such as SHAP and LIME will be incorporated in order to aid security analysts in understanding decisions, and expedite reaction activities like highlighting the file characteristics or API requests that were most important in the identification process.

References

[1] J. De Groot, "*A History of Ransomware Attack: The Biggest and Worst Ransomware Attack of All Time*," 2017 Internet Crime Report, Report, 2018. [Google Scholar] [Publisher Link]

[2] Craig Beaman et al., "Ransomware: Recent Advances, Analysis, Challenges and Future Research Directions," *Computers & Security*, vol. 111, pp. 1-22, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Umaru Adamu, and Irfan Awan, "Ransomware Prediction using Supervised Learning Algorithms," *7th International Conference on Future Internet of Things and Cloud*, Istanbul, Turkey, pp. 57-63, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Goteng Kuwunidi Job et al., "Impacts of Ransomware Attacks on Edge Computing Devices: Challenges and Research Opportunities," *International Journal of Engineering Research & Technology*, vol. 10, no. 4, pp. 665-670, 2021. [Google Scholar] [Publisher Link]

[5] Stephan Dreiseitl, and Lucila Ohno-Machado, "Logistic Regression and Artificial Neural Network Classification Models: A Methodology Review," *Journal of Biomedical Informatics*, vol. 35, no. 5-6, pp. 352-359, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Cisco Cyber Threat Trends Report: From Trojan Takeovers to Ransomware Roulette. [Online]. Available: https://umbrella.cisco.com/info/cyber-threat-trends-report

[7] A.S.S.V. Lakshmi Pooja, and M. Sridhar, "Analysis of Phishing Website Detection Using CNN and Bidirectional LSTM," *4th International Conference on Electronics, Communication and Aerospace Technology*, pp. 1620-1629, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Amjad Alraizza, and Abdulmohsen Algarni, "Ransomware Detection using Machine Learning: A Survey," *Big Data and Cognitive Computing*, vol. 7, no. 3, pp. 1-24, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Nisreen Alzahrani, and Daniyal Alghazzawi, "A Review on Android Ransomware Detection using Deep Learning Techniques," *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, pp. 330-335, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, and Syed Zainudeen Mohd Shaid, "Ransomware Threat Success Factors, Taxonomy, and Countermeasures: A Survey and Research Directions," *Computers & Security*, vol. 74, pp. 144-166, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Jinal P. Tailor, and Ashish D. Patel, "A Comprehensive Survey: Ransomware Attacks Prevention, Monitoring and Damage Control," *International Journal of Research and Scientific Innovation*, vol. 4, no. 15, pp. 116-121, 2017. [Google Scholar]

[12] Umara Urooj et al., "Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions," *Applied Sciences*, vol. 12, no. 1, pp. 1-45, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Damien Warren Fernando, Nikos Komninos, and Thomas Chen, "A Study on the Evolution of Ransomware Detection using Machine Learning and Deep Learning Techniques," *IoT*, vol. 1, no. 2, pp. 551-604, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Malak Aljabri et al., "Ransomware Detection Based on Machine Learning Using Memory Features," *Egyptian Informatics Journal*, vol. 25, pp. 1-8, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Samah Alsoghyer, and Iman Almomani, "Ransomware Detection System for Android Applications," *Electronics*, vol. 8, no. 8, pp. 1-36, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[16] Hanqi Zhang et al., "Classification of Ransomware Families with Machine Learning Based OnN-Gram of Opcodes," *Future Generation Computer Systems*, vol. 90, pp. 211-221, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] G. Kirubavathi, and W.R. Anne, "Behavioral Based Detection of Android Ransomware Using Machine Learning Techniques," *International Journal of System Assurance Engineering and Management*, vol. 15, no. 9, pp. 4404-4425, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Seong Il Bae, Gyu Bin Lee, and Eul Gyu Im, "Ransomware Detection using Machine Learning Algorithms," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Fakhroddin Noorbehbahani, Farzaneh Rasouli, and Mohammad Saberi, “Analysis of Machine Learning Techniques for Ransomware Detection,” *16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology*, Mashhad, Iran, pp. 128-133, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[20] Subash Poudyal, Kul Prasad Subedi, and Dipankar Dasgupta, “A Framework for Analyzing Ransomware using Machine Learning,” *IEEE Symposium Series on Computational Intelligence*, Bangalore, India, pp. 1692-1699, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[21] Fadare Oluwaseun Gbenga, Adetunmbi Adebayo Olusola, and Oyinloye Oghenerukevwe Elohor, “Towards Optimization of Malware Detection using Extra-Tree and Random Forest Feature Selections on Ensemble Classifiers,” *The International Journal of Recent Technology and Engineering*, vol. 9, no. 6, pp. 223-232, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[22] G. Revathy et al., “Smurf Attack Using Hybrid Machine Learning Technique,” *AIP Conference Proceedings*, vol. 2463, no. 1, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] Iram Bibi et al., “An Effective Android Ransomware Detection Through Multi-Factor Feature Filtration and Recurrent Neural Network,” *2019 UK/China Emerging Technologies*, Glasgow, UK, pp. 1-4, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[24] Maher G.M. Abdolrasol et al., “Artificial Neural Networks Based Optimization Techniques: A Review,” *Electronics*, vol. 10, no. 21, pp. 1-43, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Shun Tobiyama et al., “Malware Detection with Deep Neural Network Using Process Behavior,” *2016 IEEE 40th Annual Computer Software and Applications Conference*, Atlanta, GA, USA, pp. 577-582, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[26] Dipendra Pant, and Rabindra Bista, “Image-Based Malware Classification using Deep Convolutional Neural Network and Transfer Learning,” *Proceedings of the 3rd International Conference on Advanced Information Science and System*, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Umme Zahoora et al., “Ransomware Detection using Deep Learning Based Unsupervised Feature Extraction and a Cost Sensitive Pareto Ensemble Classifier,” *Scientific Reports*, vol. 12, no. 1, pp. 1-15, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Ransomware Detection using Machine Learning. [Online]. Available: https://github.com/muditmathur2020/RansomwareDetection/tree/master

[29] Paula Branco, “Exploring the Impact of Resampling Methods for Malware Detection,” *IEEE International Conference on Big Data*, Atlanta, GA, USA, pp. 3961-3968, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[30] Iman Almomani et al., “Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data,” *IEEE Access*, vol. 9, pp. 57674-57691, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[31] Tohari Ahmad, and Mohammad Nasrul Aziz, “Data Preprocessing and Feature Selection for Machine Learning Intrusion Detection Systems,” *ICIC Express Letters*, vol. 13, no. 2, pp. 93-101, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[32] Sumith Maniath et al., “Deep Learning LSTM based Ransomware Detection,” *2017 Recent Developments in Control, Automation & Power Engineering*, Noida, India, pp. 442-446, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[33] Amardeep Singh et al., “Enhancing Ransomware Attack Detection Using Transfer Learning and Deep Learning Ensemble Models on Cloud-Encrypted Data,” *Electronics*, vol. 12, no. 18, pp. 1-31, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[34] Bin Zhang et al., “Ransomware Classification using Patch-Based CNN and Self-Attention Network on Embedded N-Grams of Opcodes,” *Future Generation Computer Systems*, vol. 110, pp. 708-720, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[35] Revathy Ganapathy et al., “CNN-LSTM: Development of Offline Signature Authentication,” *International Conference on Emerging Research in Computational Science*, Coimbatore, India, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]