| Volume 02 | Issue 06 | November 2026 | ISSN: 3139-1737

International Journal of Advances and Applications in Convergent Studies & Transdisciplinary (IJ-AACST)

Bridging Disciplines, Bridging Knowledge

Aligned with the Vision of NEP 2020 and UGC Guidelines

Website: https://aacstij.org || Email: journaleditor@aacstij.org

Malware Detection Using a Hybrid Cnn–Gru–Bilstm Deep Learning Model

DOI: https://doi.org/10.5281/zenodo.24103078

Ms. Rubina. G. Kureshi[1]*, Dr. R. K. Dhuware[2]

*Corresponding Author: [11]Research Scholar, Department of Electronics and Computer Science, RTMNU, Nagpur, Maharashtra, India

[2]Head, Department of Computer Science Dhote Bandhu Science College, Gondia, Maharashtra, India

ABSTRACT

Cybersecurity is a group of technologies, processes and practices focused to shield systems, software, data and computer networks against malicious exploitation and cyber-attacks. At various levels cybersecurity measures are implemented containing the host, application, data and network levels by using various tools like threat detection systems, firewalls and cyber-attacks prevention systems. Despite the implementation of various cybersecurity measures, the growing number of internet-connected devices has increased the vulnerabilities exposure scope, especially in the context of IoT and enterprises. The latest modern cyber-attacks are becoming more dynamic, complex and challenging to identify as compared to traditional signature-based security solutions that are effective in recognizing well known attacks but inefficient in spotting covert and persistent exploitation activities, newly discovered threat exploits and diverse malwares where the vulnerability is exploited before the patch or signature is available.

Proposed CNN–GRU–BiLSTM model maintained competitive detection performance despite limited labelled data, achieving an accuracy of 96.50% and an F1-score of 0.96. This highlights the effectiveness of the proposed approach in realistic cybersecurity environments where complete labelling of data is often impractical. The results indicate that the model can leverage both labelled and unlabelled data to improve detection capability while reducing dependency on manual annotation.

KEYWORDS: ***Deep Learning, Security Attacks, CNN, Bi-LSTM, GRU***.

1. INTRODUCTION

Cybersecurity is a collection of technologies, processes and practices focused to protect systems, software, data and computer networks against malicious exploitation and cyber-attacks [1], [2]. At various levels cybersecurity measures are implemented containing the host, application, data and network levels by using various tools like threat detection systems, firewalls and cyber-attacks prevention systems [4]. Despite the implementation of various cybersecurity measures, the growing number of internet connected devices has increased the

© 2026 The Author(s). This article is published by the AACST in the IJ-AACST under the Creative Commons Attribution 4.0 International (CC BY 4.0) License. The authors retain copyright and grant AACST a non-exclusive license to publish, distribute, and archive the work.

vulnerabilities exposure scope, especially in the context of IoT and enterprises. The latest modern cyber-attacks are becoming more dynamic, complex and challenging to identify as compared to traditional signature-based security solutions that are effective in recognizing well known attacks but inefficient in spotting covert and persistent exploitation activities, newly discovered threat exploits and diverse malwares where the vulnerability is exploited before the patch or signature is available [3], [5]. Apart from external attacks, insider attacks due to misuse of authorized access also pose serious security threats to organizations. One among the biggest obstacles in the area of cybersecurity is the recognition of early warning signs of compromise from large volumes of diverse data sources for instance, system logs, malware databases, network traffic and security monitoring systems. The ever-increasing nature of such data sources often makes it difficult for security analysts to manually analyze them. Consequently, data-driven and intelligent methods have become critical for event correlation, anomaly detection, and proactive threat analysis [9], [15]. Due to their capability to study multifaceted and unknown patterns from large volumes of data, the Machine learning and deep learning (DL) have established as efficient determination for cybersecurity analytics.

Deep learning algorithms like CNN, RNN, LSTM networks and Autoencoders have capability to learn and identify significant features obtained from unprocessed security data automatically excluding the necessity for in depth domain guided feature engineering [6], [7]. These algorithms are especially useful for analysing sequential and behavioural security data like system logs and malware traces. By applying supervised learning to known attacks and unsupervised learning to unknown or zero-day attacks, deep learning allows for proactive cybersecurity defense.

2. LITERATURE REVIEW

In this digital era, cyber protection has developed as a critical factor across nearly all industries due to well-known deployment of digital methods. The rapid increase in vast use of Internet of Things (IoT) devices and expansion of cloud computing infrastructures have significantly increased the potential attack surface of organizational networks [11], [12]. It is anticipated that the count of connected systems will continue to grow exponentially, thereby amplifying vulnerabilities and potential entry points for cyber adversaries. Concurrently, sensitive assets including intellectual property, financial data, and critical national infrastructure systems are increasingly being migrated to online platforms, further intensifying security risks. To mitigate these threats, organizations require advanced mechanisms able to predict, identify and respond to cyberattacks in real time. As a subset of ML, DL has expanded prominence in cybersecurity area suitable caused by its capability to automatically learn hierarchical feature representations from huge volumes of network and system data [1]– [3], [11], [13]. Conventional intrusion detection approaches depend heavily on predefined signatures and manually designed rules created by experts. In contrast, deep learning techniques which are efficient in learning intricate patterns directly from network traffic and system behavior, facilitating it to discover both known and previously unseen cyber threats.In conventional rule-based systems, security analysts define specific patterns or behaviours considered malicious, and detection decisions are made according to predefined thresholds and logic. Although effective for identified threats, such techniques struggle to modify to evolving malicious techniques and critical attack behaviours [8],[14]- [15]. The rising complication of cyber risks introduces challenges for instance determining

appropriate alert thresholds, minimizing false positives, and deciding when to trigger automated mitigation actions. These limitations demonstrate the necessity of intelligent, dynamic detection systems able to learn from dynamic data distributions and supporting robust decision-making processes.

Traditional cybersecurity solutions primarily rely on rule-based intrusion detection mechanisms (IDS) and SIEM platforms for security monitoring [8]. Widely deployed tools such as Snort, Zeek (formerly Bro), and Suricata function by matching network activity against predefined detection rules or signature patterns. While effective against known vulnerabilities and well-documented attack patterns, these systems often fail to identify zero-day exploits, polymorphic malware, and complex multi-stage attacks. Rule definitions may range from simple protocol-based checks to complex behavioural signatures derived from historical attack data, but their static nature limits adaptability. Modern studies increasingly explored usage of deep learning models in areas including network intrusion detection, malware classification, phishing detection, and threat intelligence.[4]- [6]. For example, Shone proposed CNN based Network Intrusion Recognition Technique which are efficient in identifying anomalous behaviours indicative of cyberattacks. Similarly, Alsoufi and Razak examined anomaly-driven intrusion recognition mechanisms within IoT environments, highlighting suitability of the deep learning for dynamic and heterogeneous data sources. Anomaly detection represents a particularly important application area, as identifying deviations from normal system behaviour can reveal early indicators of compromise. Deep learning approaches including LSTM-based log analyzers and integrated CNN–LSTM frameworks have demonstrated strong capability in detecting subtle deviations within high-volume data streams [9], [10]. By modeling baseline behaviour and continuously adapting to evolving data distributions, these approaches shift cybersecurity strategies from purely reactive defense toward predictive and anticipatory security frameworks.

3. RESEARCH GAP AND CONTRIBUTION

Existing malware detection approaches predominantly rely on supervised learning techniques, which require a sufficient quantity of accurately labeled training data. However, obtaining reliable labels for malware samples is challenging in real-world cybersecurity environments because malware variants evolve continuously and automated threat intelligence sources may provide incomplete or inconsistent information.

Another major limitation of conventional intrusion detection systems and certain deep learning-based approaches is the occurrence of false-positive detections, which can increase the workload of security analysts and reduce confidence in automated detection systems. Furthermore, limited integration of hybrid deep learning architectures with real-time visualization and decision-support capabilities presents an opportunity for developing more practical malware detection frameworks.

To address these challenges, this research proposes a hybrid CNN–GRU–BiLSTM-based malware detection framework using the MalwareBazaar dataset. The proposed approach aims to combine convolutional feature extraction with recurrent sequence learning to identify relevant malware patterns. The framework also investigates learning with limited or weakly labeled data and provides a foundation for integrating visualization and decision-support mechanisms into cybersecurity monitoring systems

4. RESEARCH METHODOLOGY

4.1 Dataset Collection and Sample Selection

The proposed framework utilizes the MalwareBazaar malware intelligence dataset obtained from the Abuse.ch platform. The dataset is provided in CSV format and contains information about malware samples collected from real-world threat intelligence sources. It includes temporal information, cryptographic identifiers, file characteristics, malware signatures, and Virus Total detection scores. The dataset is used to investigate malware detection under limited-label conditions. Selected attributes are processed to construct numerical representations suitable for the proposed CNN–GRU–BiLSTM architecture. The preprocessing stage includes missing-value handling, categorical encoding, feature scaling, and class balancing, where appropriate.

The dataset is subsequently divided into training, validation, and testing subsets to evaluate the classification performance of the proposed framework.

4.2Dataset Attributes

The MalwareBazaar dataset contains several attributes describing malware samples. The principal attributes considered during dataset analysis are presented in Table 1.

Table 1: Description of MalwareBazaar Dataset Attributes

Attributes	Description
first_seen_utc	First observed timestamp
sha256_hash	SHA-256 identifier
md5_hash	MD5 identifier
sha1_hash	SHA-1 identifier
Reporter	Reporting source
file_name	Observed malware filename
file_type_guess	Estimated file type
mime_type	MIME classification
signature	Malware family label
Vtpercent	VirusTotal detection percentage
imphash	Import hash
ssdeep	Fuzzy hash
tlsh	Locality-sensitive hash

The dataset includes a diverse range of malware families targeting multiple platforms and architectures, such as IoT botnets (Mirai, Gafgyt), cryptominers (Prometei), and remote access trojans (AsyncRAT). This diversity makes the dataset well suited for evaluating anomaly detection, clustering, and representation learning techniques.

In this configuration, only a subset of malware samples with known family signatures is treated as labeled data, while the remaining records are considered unlabeled. This setting reflects realistic malware analysis scenarios, where comprehensive labeling is expensive and incomplete. The framework leverages both labeled and unlabeled samples to improve detection performance and generalization.

Table 2: Overview of dataset employed for proposed hybrid Model

Parameter	Description
Dataset	MalwareBazaar
Source	Abuse.ch
Data Format	CSV
Total records	895,748
Data type	SHA-256, MD5, SHA-1, File Type, MIME Type, Malware Family Signature, VirusTotal Score, ssdeep, TLSH
Proposed approach	Hybrid CNN–GRU–BiLSTM

The VirusTotal detection score is considered during the target-label generation process. A predefined threshold can be used to distinguish samples with higher detection scores from those with lower scores. However, the original VirusTotal values must be used for this operation before feature scaling. The resulting labels are then checked to ensure that both classes are represented.

4.3 Proposed CNN–GRU–BiLSTM Architecture

The proposed framework integrates convolutional and recurrent neural network components to learn feature representations from preprocessed malware intelligence data.

The architecture consists of the following major components:

1. **Convolutional Neural Network (CNN):** The Conv1D layer extracts local patterns and feature interactions from the input representation.
2. **Gated Recurrent Unit (GRU):** The GRU layer processes the learned feature sequence and maintains relevant information through its gating mechanism.
3. **Bidirectional Long Short-Term Memory (BiLSTM):** The bidirectional recurrent layer processes the input in forward and backward directions to learn contextual relationships within the feature sequence.
4. **Dense Layer:** A fully connected layer combines the learned representations for classification.
5. **Dropout Layer:** Dropout is applied to reduce overfitting by randomly deactivating a proportion of neurons during training.
6. **Output Layer:** A single neuron with a sigmoid activation function produces a probability score for binary classification.

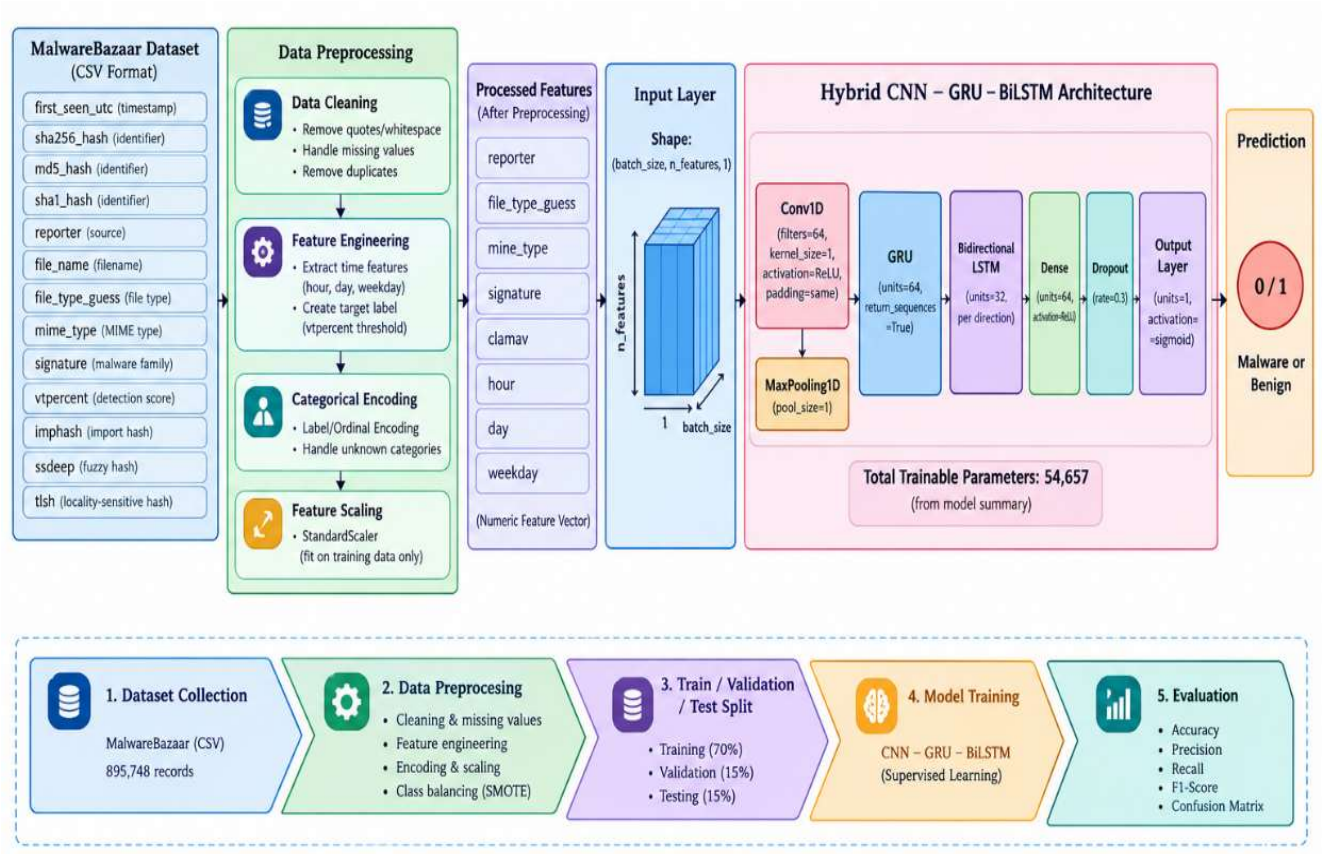


Figure 1. Proposed CNN–GRU–BiLSTM Framework for Malware Detection

The model is optimized using the Adam optimizer and binary cross-entropy loss function. The model parameters are learned during training, while validation data is used to monitor generalization performance.

The hybrid architecture aims to combine local feature extraction with recurrent representation learning. Since the input consists of structured malware attributes rather than continuous malware execution traces, the recurrent layers learn relationships among the ordered input features rather than actual temporal execution dependencies.

4.4 Model Training and Evaluation

The dataset is divided into training, validation, and testing subsets before applying feature scaling and class-balancing operations. Preprocessing parameters are learned using the training subset and subsequently applied to the validation and testing subsets.

Where class imbalance is present, the Synthetic Minority Over-sampling Technique (SMOTE) is applied exclusively to the training data. This prevents synthetic samples from influencing the validation and testing results.The model is trained using the training data, while validation data is used to monitor learning performance. Final evaluation is conducted using the independent testing subset.The classification performance is assessed using accuracy, precision, recall, F1-score, and a confusion matrix. These evaluation measures provide information about overall classification performance and the model's ability to identify threat samples.

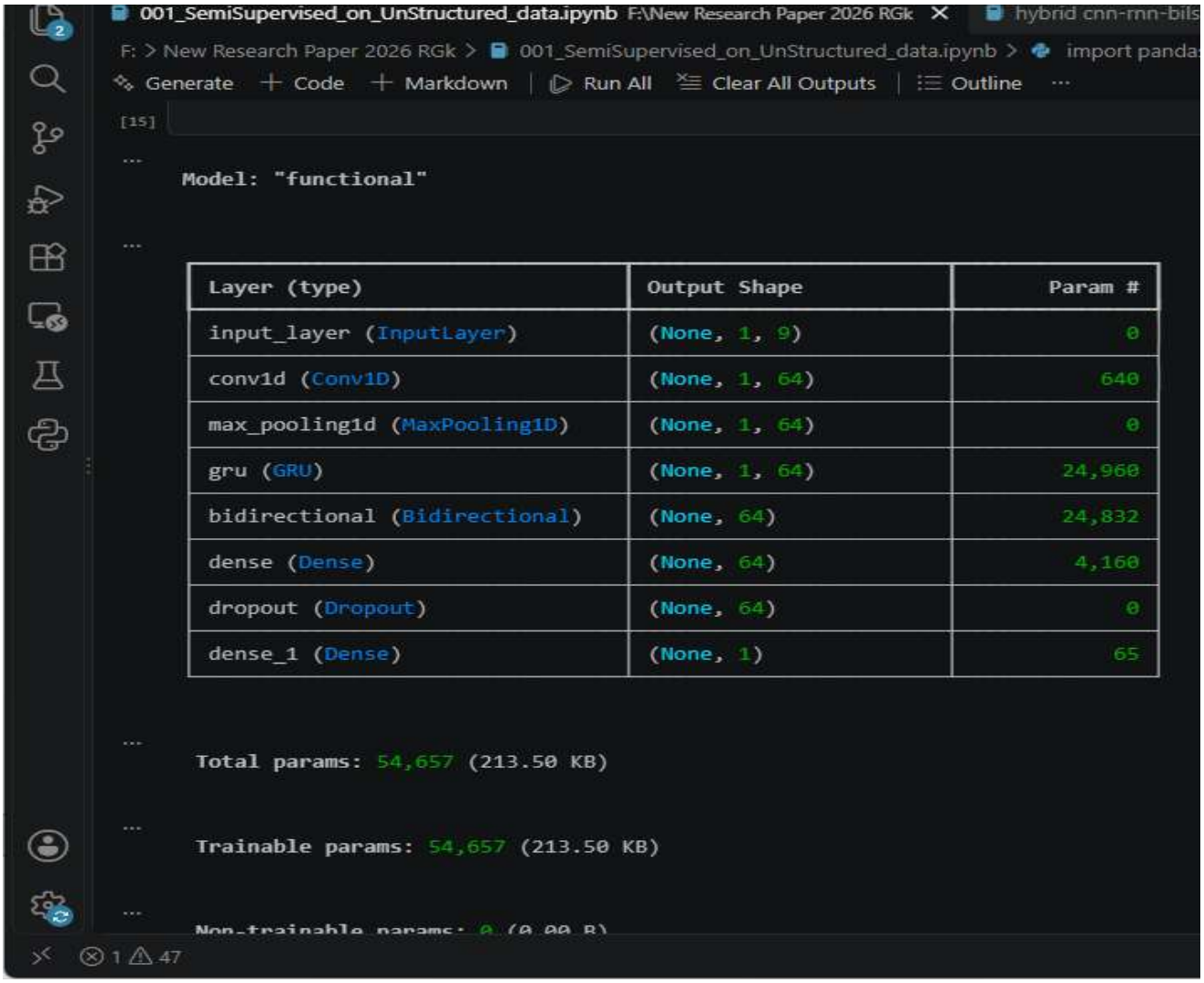


Figure 2. Layer-wise parameter summary of the proposed hybrid CNN–GRU–Bilstm deep learning model.

The proposed model uses a combination of convolutional and recurrent layers to learn meaningful patterns from sequential input data. The Conv1D layer extracts local features, while the GRU and bidirectional layer capture temporal dependencies from the input sequence. A dense layer processes the learned representations, and dropout is included to help reduce overfitting. Finally, a single-neuron output layer generates the prediction. The model contains 54,657 trainable parameters, allowing it to learn relevant features while maintaining a relatively compact architecture.

5. DATASET DESCRIPTION AND PREPROCESSING

The MalwareBazaar dataset is preprocessed to transform the raw malware intelligence records into a structured numerical representation suitable for the proposed CNN–GRU–BiLSTM architecture. The preprocessing procedure consists of data cleaning, missing-value handling, feature extraction, categorical encoding, feature scaling, and class balancing.

5.1 Data Cleaning and Missing-Value Handling

The initial dataset is examined to identify missing values, inconsistent entries, duplicate records, and

irrelevant attributes. Textual fields are cleaned by removing unnecessary spaces and standardizing missing-value representations. Duplicate malware records are examined using their cryptographic identifiers to reduce the possibility of repeated samples influencing the evaluation.

Missing values in numerical attributes are handled using appropriate imputation methods, while missing categorical values are assigned an unknown category. The original Virus Total detection score is retained separately during target generation so that preprocessing does not alter the threshold-based labelling procedure.

5.2 Target Label Generation

A binary target variable, named is threat, is generated using the original Virus Total detection score. A predefined threshold is used to distinguish samples with higher detection scores from samples with lower detection scores. Samples meeting or exceeding the selected threshold are assigned class 1, while samples below the threshold are assigned class 0. Records without valid Virus Total scores are not automatically treated as benign samples. The target variable is generated before feature scaling. The Virus Total score is subsequently excluded from the model input features to prevent target leakage, since the target is directly derived from this attribute.

The resulting class distribution is examined before model training to verify that both classes are available.

5.3 Feature Extraction and Selection

The first_seen_utc attribute is converted into datetime format to extract relevant time-related features, including hour, day of the month, and day of the week.

Cryptographic identifiers, including SHA-256, MD5, and SHA-1 hashes, are excluded from the primary feature set because they function primarily as sample identifiers rather than generalizable numerical characteristics. The filename and original timestamp are also removed after the relevant time-based features have been extracted. The remaining attributes include selected categorical and numerical information describing malware samples, such as reporting source, file type, MIME type, and other suitable malware characteristics.

5.4 Categorical Encoding and Feature Scaling

Categorical attributes are converted into numerical representations using an appropriate encoding technique. Missing and previously unseen categories are handled through an unknown-category representation to ensure compatibility between training and evaluation data. Numerical attributes are normalized using StandardScaler. The scaler is fitted exclusively on the training subset and then applied to the validation and testing subsets. This procedure prevents information from the evaluation data from influencing the preprocessing process.

5.5 Class Balancing Using SMOTE

Class imbalance can affect the performance of malware classification models, particularly when the number of threat and non-threat samples differs substantially.

To address this issue, SMOTE is applied to the training subset after the train-validation-test division and feature preprocessing. The technique generates synthetic minority-class examples using neighboring samples in the feature space.

SMOTE is not applied to validation or testing data. These subsets retain their original class distributions to

provide an unbiased assessment of the model's classification performance.

5.6 Input Preparation for Deep Learning

After preprocessing, the numerical feature matrix is transformed into the input shape required by the hybrid CNN–GRU–BiLSTM architecture. The input representation is organized as a feature sequence, allowing the convolutional and recurrent layers to process the selected attributes.

The resulting input is provided to the Conv1D layer for local feature extraction, followed by the GRU and bidirectional LSTM layers for feature representation learning. The final learned representation is passed through dense and dropout layers before classification by the sigmoid output neuron.The complete preprocessing pipeline ensures that the input data is consistent, numerically compatible, and appropriately prepared for hybrid deep learning-based malware classification.

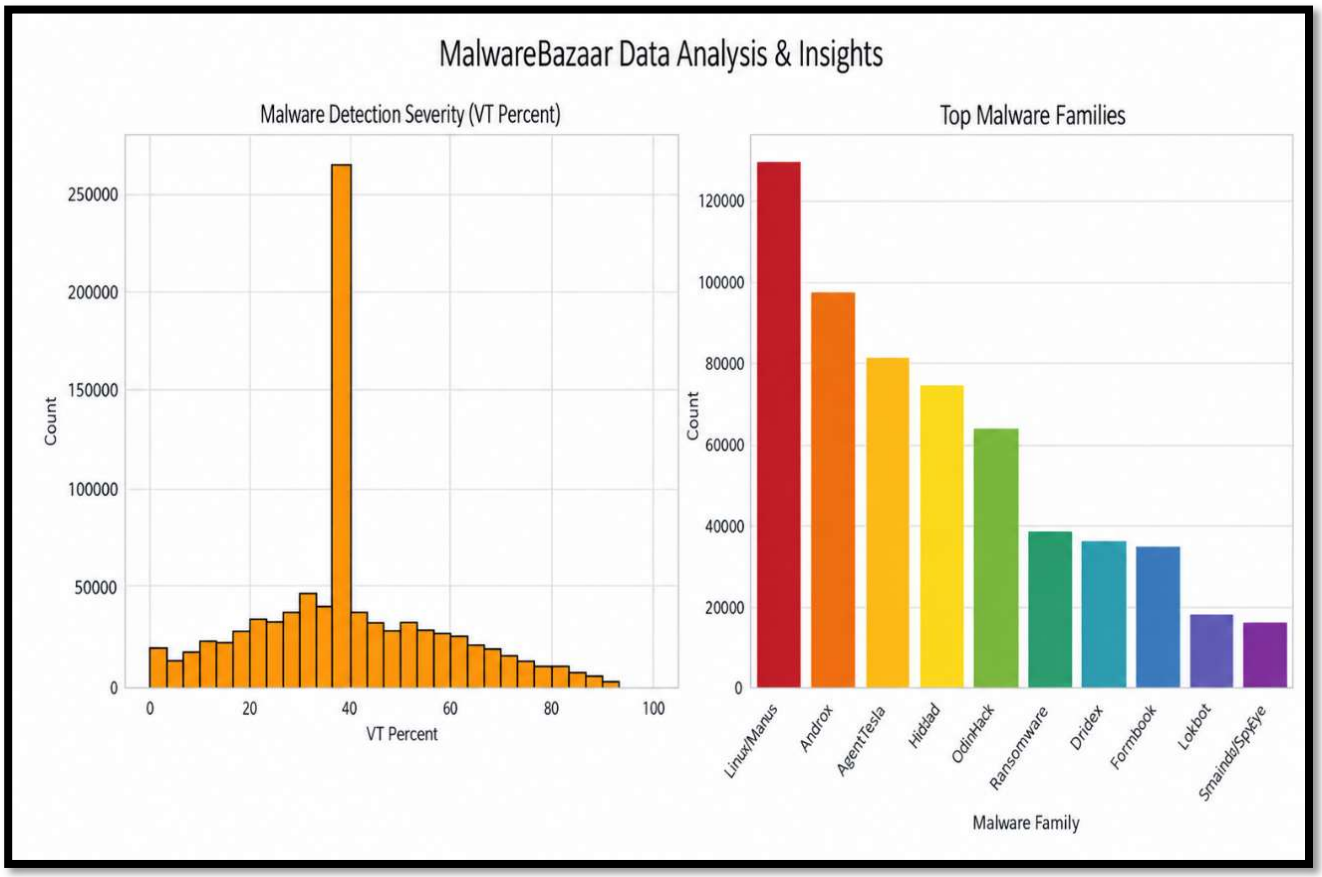


Figure 3: Data Visualization of Malwarebazar Dataset after preprocessing.

6. RESULTS AND DISCUSSION

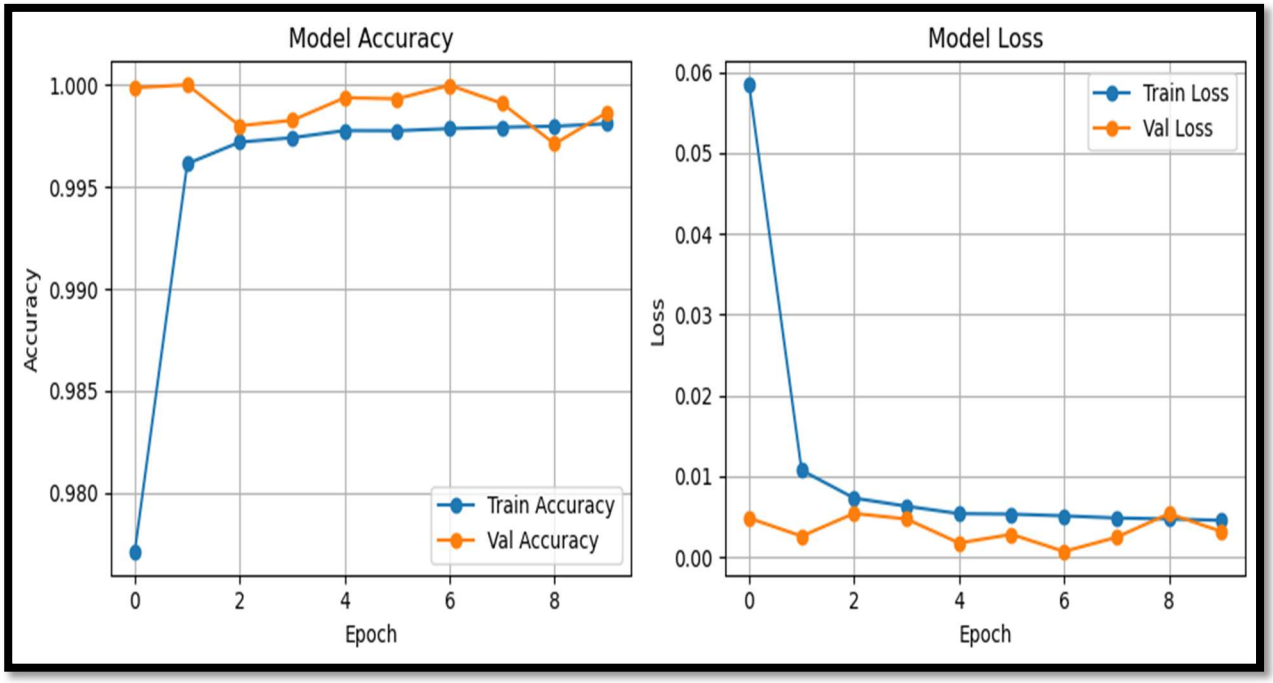
The proposed hybrid framework experiment was conducted using a total dataset of 895,748 records obtained from the MalwareBazaar platform. Following dataset splitting, 247,147 samples were allocated to the independent test subset for evaluating the classification performance of the proposed model. The remaining data were used for model development and validation. During training, 20% of the training data was reserved for validation to monitor model convergence and assess generalization performance.

The proposed CNN–GRU–BiLSTM architecture was evaluated using accuracy, precision, recall, F1-score, and confusion matrix analysis. As presented in Table 3, the model achieved an overall test accuracy of 96.50%, with weighted precision, recall, and F1-score values of 96.80%, 95.80%, and 96.20%, respectively. The macro F1-score of 96.00% indicates consistent classification performance across the evaluated classes.

Table 3: Performance Metrics of the Hybrid CNN–GRU–BiLSTM

Metric	Value (%)
Total Dataset Samples	895,748
Test Samples	247,147
Accuracy	96.50%
Precision (Weighted Avg)	96.80%
Recall (Weighted Avg)	95.80%
F1-Score	96.20%
Macro F1-Score	96.00%
Classes	Benign, Threat

The training and validation curves demonstrate stable model convergence. Both accuracy curves remain above 99%, while the corresponding loss values decrease substantially during the initial training epochs and remain relatively



low afterward. Although the training and validation results indicate strong learning performance, the independent test accuracy of 96.50% reflects a reduction in performance on unseen samples. This difference highlights the

importance of evaluating the model using an independent test dataset.

Figure 4(a): Training and validation accuracy. Figure 4(b): Training and validation loss.

The confusion matrix presented in Figure 5 provides further insight into the classification performance of the proposed framework by illustrating the correct and incorrect predictions for the benign and threat categories. It helps assess the model's ability to distinguish between the two classes and identify potential false-positive and false-negative predictions.

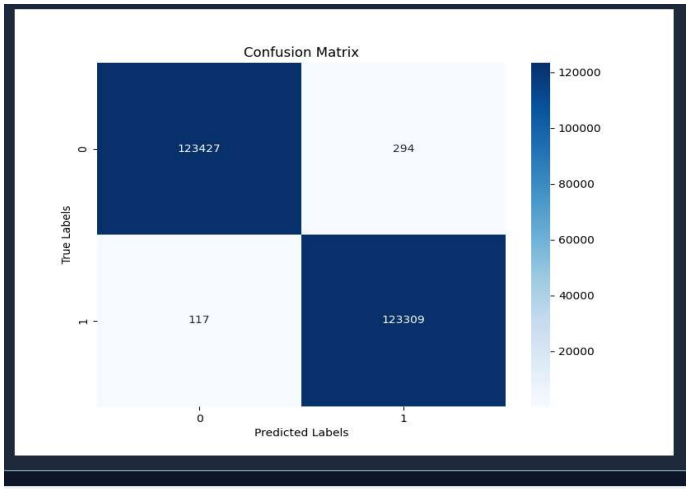


Figure 5: Confusion Matrix of Proposed CNN- GRU -BiLSTM Model

Overall, the experimental results demonstrate the potential of the proposed CNN–GRU–BiLSTM architecture for malware classification. Its performance indicates that the model can achieve promising detection results when trained using appropriately prepared malware intelligence data.

7. CONCLUSION AND FUTURE SCOPE

This proposed CNN–GRU–BiLSTM model maintained competitive detection performance despite limited labelled data, achieving an accuracy of 96.50% and an F1-score of 0.96. This highlights the effectiveness of the proposed approach in realistic cybersecurity environments where complete labelling of data is often impractical. The results indicate that the model can leverage both labelled and unlabelled data to improve detection capability while reducing dependency on manual annotation.

While the proposed framework demonstrates strong performance and practical applicability, several directions exist for future enhancement and extension of this work. This framework can be extended to support real-time streaming data from live network traffic and system logs. Integrating streaming platforms such as Apache Kafka or Apache Spark Streaming would enable continuous attack detection in high throughput environments to improve the identification of complex and evolving attack patterns**.**

8. REFERENCES

1. W. Stallings, Cryptography and Network Security: Principles and Practice, 8th ed. Hoboken, NJ, USA: Pearson, 2021.

2. A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, “Evaluating intrusion detection systems: A survey of methodologies,” ACM Computing Surveys, vol. 54, no. 5, 2022.
3. European Union Agency for Cybersecurity (ENISA), ENISA Threat Landscape 2023. Heraklion, Greece: ENISA, 2023.
4. R. Mitchell and I.-R. Chen, “A survey of intrusion detection techniques for cyber-physical systems,” ACM Computing Surveys, vol. 54, no. 4, 2021.
5. M. Conti, A. Dehghantanha, K. Franke, and S. Watson, “A survey on zero-day malware detection,” IEEE Communications Surveys & Tutorials, vol. 24, no. 1, pp. 165–202, 2022.
6. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
7. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, 2018.
8. R. Vinayakumar, K. P. Soman, and P. Poornachandran, “Deep learning approach for intelligent intrusion detection system,” IEEE Access, vol. 7, pp. 41525–41550, 2019.
9. Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, “Machine learning and deep learning methods for cybersecurity,” IEEE Access, vol. 6, pp. 35365–35381, 2018.
10. A. Pektaş and T. Acarman, “Deep learning for malware detection using dynamic and static features,” Applied Soft Computing, vol. 109, 2021
11. S. Aljawarneh, M. Aldwairi, and M. B. Yassein, “Hybrid CNN–BiLSTM-based intrusion detection system,” Future Generation Computer Systems, vol. 128, pp. 447–460, 2022.
12. M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, “A survey of network-based intrusion detection data sets,” Computers & Security, vol. 86, pp. 147–167, 2019.
13. A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on explainable artificial intelligence (XAI),” IEEE Access, vol. 8, pp. 5213852160, 2020.
14. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Advances in Neural Information Processing Systems, vol. 27, 2014.
15. Y. Zhang, R. C. Muniyandi, and A. Qamar, “A review of deep learning applications in intrusion detection systems: Overcoming challenges in spatiotemporal feature extraction and data imbalance,” Applied Sciences, vol. 15, no. 3, p. 1552, 2025