Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

AI-Driven Malware Detection and Prevention using Hybrid Machine Learning and Blockchain for Secure Cyber Threat Intelligence

Bharti Ahuja Salunke[1], Sharad Salunke[2]

Department of CSE, Poornima University, Jaipur, India.

Email: [1]bharti.salunke99@gmail.com, [2]sharad.sal@gmail.com

Abstract

The ever-evolving cyber threat landscape has enabled sophisticated and intelligent malware detection techniques. False positive rates and inadequate attack pattern adaptation are common limitations of conventional intrusion detection algorithms. In an effort to improve security, this work offers a combined malware detection method based on artificial intelligence. Additionally, it integrates deep learning and machine learning with blockchain technology. By using Random Forest and Long Short-Term Memory for feature selection and anomaly detection, the suggested system can identify cyber threats with greater accuracy. A blockchain ledger also facilitates the recording of attack indicators, enhancing threat intelligence. The proposed method outperformed the standalone ML/DL results with 99.7% accuracy, 99.5% precision, 99.4% recall, and a 99.5% F1-score on the chosen dataset. Blockchain technology further enhances incremental trust in security by ensuring confidence in cybersecurity agencies by eliminating data manipulation.

Keywords: Cybersecurity, Malware Detection, Random Forest, Machine Learning, Blockchain.

1. Introduction

With the growth of advanced technologies like AI, cyber threats have also increased. Therefore, in the context of cybersecurity, the importance of malware detection has risen. To safeguard the data of various areas, such as individual, organizational, and governmental, from malicious software like viruses, worms, trojan horses, ransomware, and spyware, traditional methods are inadequate because they rely on signature- and heuristic-based techniques. In the signature-based method, files are compared to known malware signatures for identification. Although this method is effective, it fails to identify zero-day attacks and polymorphic malware because these adapt their variable code to avoid detection [1]. On the other hand, heuristic-based detection is efficient but has a high false positive rate, often flagging normal applications as potential threats.

Advanced solutions may overcome these problems through the use of machine learning-based malware detection. The ML technique, instead of relying on a set of rules, works on huge datasets to find trends. By learning from previous attacks, this technique can identify new threats. Some important methods include Support Vector Machines (SVM), deep learning

DOI: https://doi.org/10.36548/jtcsst.2025.3.015
Received: 24.07.2025, received in revised form: 28.08.2025, accepted: 15.09.2025, published: 26.09.2025

models like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), Decision Trees, and Random Forests [2]. By continuously observing traffic, system logs, API requests, and file execution patterns, these models can identify different and unusual behaviors that may lead to cyber-attacks. Unusual system behaviors may result in covert malware attacks or advanced persistent threats (APT). These outliers can be detected through unsupervised learning techniques like clustering and anomaly detection.

More recent studies have shown that LSTM integration can improve the precision and effectiveness of existing deep learning frameworks. One study found that CNN and LSTM are effective in detecting malware, classifying binary images, and classifying opcode-sequences [3]. Similarly, for malware intrusion detection, a CNN + LSTM + attention mechanism can provide low latency and high accuracy [4]. When labeled data is not available, anomalies can be found using autoencoders and Deep Belief Networks (DBNs). Additionally, evasive or zero-day malware can be detected by GAN-based frameworks like AED-GAN [5]. This framework increases detection efficacy, decreases false positives, and enhances adaptability in the face of the cyberthreats.

Denial of service (DoS) attacks, botnets, and phishing are just a few of the business and cybersecurity domains where the use of machine learning-based malware detection platforms can be beneficial. The goal of intrusion detection and prevention systems (IDPS) is to lessen these kinds of attacks. Machine learning, which processes network data in real time, can help achieve IDPS. Next-generation antivirus software, which is used to identify and categorize malicious code, can be developed using machine learning models. The cloud security space can handle malicious API requests and unauthorized access attempts thanks to the convergence of cloud technology and machine learning. ML models can play a key role in identifying fraud in the era of cybercrime, which includes banking and financial transactions. Security information and event management (SIEM) systems can also use machine learning techniques to manage cyber incidents and measure threat intelligence data. [6] In spite of the major advantages of ML in malware detection, there are also some problems too with real-world instances:

1. Modification of data by fraudsters to confuse machine learning models, for example, identifying malware as benign [7].
2. Deep learning models demand large amounts of memory and power, that is, scalability, for analyzing massive cybersecurity data [5], [10].
3. False positives and false negatives: incorrect identification may overburden the system with false warnings or fail to notice a malware attack.

To address these stated issues, researchers are finding solutions in the domain of explicable artificial intelligence (XAI) models. They provide transparency in decision-making, ensuring that teams know the accurate reasons for the activation of alarms. In malware detection, AI using blockchain is the most recent development in the field. Blockchain provides a secure, tamper-proof platform for storing attack logs, which increases security capabilities [11]. Data-changing activity can be stopped with the help of blockchain stored malware and signature detection records. It also enables data sharing among various agencies. With this activity, trust among security agencies would be increased, and they would be encouraged to make joint defensive measures against cyberattacks.

With the advancement and complexity of attacks, technology that integrates various facets, such as ML, DL, and blockchain, may provide incremental security against malware attacks. Real-time cyber-threat detection is important in the realm of financial and banking operations for cloud-based services. Therefore, AI-driven techniques should have the capability of detect cyberattacks. Intelligent cybersecurity solutions are adopted by various organizations to maintain an advantage over hackers and for malware protection measures. In this work, a number of important contributions are delivered, such as RF for feature selection and LSTM for deep anomaly detection; this provides an improvement in classification accuracy. Other contributions are mentioned below:

1. Prevention of signature alteration and hindrance of its tracking through blockchain-integrated threat logging.
2. Reduction in false positives with the proposed hybrid approach compared to the standard intrusion detection system (IDS).
3. Centralized failure can be eliminated with blockchain technology, which can also enable security node validation and secure exchange of attack intelligence.
4. Detection accuracy of 99.7% with the CIC-IDS2017 dataset, showing its superiority over independent ML & DL models.

2. Related Work

- For malware detection, conventional techniques primarily rely on heuristic and signature-based methods.
- Signature-based Method: (a) Identifies each known malware strain. (b) Struggles to identify novel or polymorphic malware.
- Heuristic Analysis: (a) Analyzes code behavior and identifies previously unidentified malware. (b) More adaptable than the signature-based method. (c) Can produce false positives and classify software as malware due to similar behavior.

In addition to the above discussion on the two primary methods, it is noted that as the complexity of malware increases, conventional approaches become unsuitable, necessitating advanced procedures and methods. Two noteworthy tools against cyber threats are AI and ML. AI-ML has the advantage of learning from data, and learning depends on the volume of data so that the model can generate various predictions and forecasts [12], [13]. Various ML techniques have previously been applied; each technique has its own advantages.

- **Random Forest:** Generates several decision trees and integrates them to improve accuracy. RF performance is high in separating benign from harmful software by examining characteristics like file behavior, network activity, and metadata.
- **SVM:** Categorizes data into multiple groups for optimization. It is effective at differentiating benign from hostile actions; therefore, it is considered superior in accuracy.

- **Deep Learning Models:** Used to assess large datasets that consist of file contents and behavior patterns. This can evolve over time by learning complex patterns, therefore performing better in identifying previously unidentified malware.

Table 1 highlights the key differences between traditional malware detection techniques and machine learning-based malware detection methods.

Table 1. Traditional Malware Detection Vs Machine Learning-Based Malware Detection

Feature	Traditional Malware Detection	Machine Learning-Based Malware Detection
Detection Approach	Signature-based, rule-based heuristics	Behavioral analysis and pattern recognition
Zero-Day Attack Detection	× Limited (Fails to detect unknown malware)	✓ Effective (Detects new & evolving threats)
Adaptability	× Needs manual signature updates	✓ Learns and updates dynamically
False Positives	High due to static rule sets	Lower due to better behavior analysis
False Negatives	High	Lower
Response Time	Slower, requires frequent database updates	Faster, real-time detection and mitigation
Computational Overhead	Low, but requires constant updates	Higher, but scalable with cloud resources
Anomaly Detection	× Not effective	✓ Excellent at spotting unusual behaviors
Resource Consumption	Low, but limited detection capabilities	Moderate to high, but better accuracy
Effectiveness Against Polymorphic Malware	× Weak (Signature-based methods fail)	✓ Strong (Learns malware behaviors dynamically)
Scalability	Limited to predefined signatures	Highly scalable with automated learning

Explainability	Transparent	May require Explainable AI
Blockchain Integration	× Not possible	✓ Can be integrated for secure logging

Unlike cryptocurrency, blockchain technology also has penetration in many other fields, such as cybersecurity. Its main features are decentralization, transparency, and immutability, which make it beneficial for protecting digital systems. In cybersecurity, blockchain is used to manage identities, protect data flow, and, in recent times, store and check malware detection results. Smart contracts are agreements that run automatically when the given conditions are met. They operate on the blockchain and perform the actions specified in the code. In malware detection, smart contracts can store detection results automatically, which keeps the records safe and prevents any changes to them. Despite all the benefits of blockchain, it has some pitfalls in cybersecurity, such as:

- Scalability issues; with the increased network size, transaction speed may lag.
- Some blockchain networks use consensus methods like Proof of Work (PoW) that require a lot of energy, raising concerns about sustainability in the context of energy consumption.
- The distributed nature of blockchain may complicate regulation and legal control in handling legal issues.

Blockchain and artificial intelligence (AI) together can provide a robust solution for better cybersecurity and malware detection. Blockchain can store and verify data securely, while AI can analyze and learn from large amounts of data. Together, they can help detect, record, and manage cyberattacks effectively.

Ansar et al. [14] study blockchain-based methods for detecting and preventing phishing, malware, insider threats, ransomware, and data breaches. These methods aim to address cybersecurity problems. Traditional detection methods can be supported by the trustless design, immutability, and decentralization features of blockchain. This study classifies current blockchain-based intrusion detection systems by type of attack, platform used, consensus mechanism, and smart contract application. It also describes the advantages and disadvantages of each group in clear and simple terms. Issues related to scalability, legal regulations, connection obstacles, and high resource consumption may be effectively addressed by blockchain. The study concludes that combining AI-based anomaly detection with blockchain can improve threat intelligence, real-time monitoring, and data security, making blockchain a powerful tool for modern cybersecurity. In this work, Alomari et al. [15] applied deep learning with correlation-based feature selection for rapid malware detection, yielding better and faster performance. The study addresses the problem of high-dimensional data in malware detection. Feature selection methods were tested on two malware datasets to reduce features while maintaining good detection results. Dense and LSTM-based deep models were used to detect normal activities and malware. Results show that some selection methods maintain high accuracy while reducing computing costs. Correlation-based selection can cut features by about 93.5% with a small loss in accuracy, demonstrating its applicability in real cybersecurity applications.

For automatic malware detection in the Android application area, Poornima et al. [16] proposed an ML & DL-based solution. It employs MAD-NET (Malware Attack Detection Network), which incorporates Deep Belief Networks (DBN) to categorize apps into harmful and non-harmful groups. On the dataset used (CICIDS2017), it generates behavior- and signature-based elements, which are further investigated by the deep learning model. Among ANN, GAN, LSTM, and DBN, DBN achieved the highest reported accuracy of 99%. Comparative analysis shows MAD-NET to be superior in detection. The study reveals that the combination of deep learning and feature engineering can effectively identify malware on Android devices, leading to reliable and scalable security. For improving intrusion detection, Abubakar et al. [17] suggested a blockchain technique with ML and AI. The method targets the issue of high false positives in conventional IDS due to limited training data and threshold settings. It uses a fusion model combining multiple IDS algorithms with weighted voting, placed inside a blockchain-secured framework. This decentralized design supports real-time monitoring, keeps attack records tamper-proof, and increases scalability. The system is tested on the DARPA 99 and MIT-Lincoln Labs datasets, achieving 92.6% accuracy and reducing false positives to 7.4%. The outcomes imply that blockchain-based IDS are more secure, reliable, and efficient in detecting threats, leading to strong and self-sustained cybersecurity solutions.

3. Methodology

To improve cybersecurity, the proposed model combines LSTM, Random Forest, and blockchain technology. The process begins with dataset collection. All the collected CICI-IDS2017 datasets are then moved to the pre-processing step. In this process, duplicates are removed, missing values are filled in and categorical data is transformed into numerical form. For the feature selection process, the model uses the Random Forest method. The model increases detection accuracy and reduces computational complexity by eliminating irrelevant information and choosing the most relevant features. An AI-based malware detection method captures the advantages of RF and LSTM.

- RF → Capturing potential threats, acts as a primary classifier.
- LSTM → Reduces complex attacks and vulnerabilities.

The two findings are combined using an ensemble technique to increase detection accuracy and lower false positives. When malware is found, SHA-256 is used to hash data like the IP address, timestamp, and attack type. The previously mentioned hashed data is stored on the blockchain network with smart contracts for security. This technology makes it possible to exchange threat intelligence in real-time against cyberattacks and store documents safely, as shown in Figure 1.

3.1 Random Forest

One example of a supervised machine learning model that carries out feature selection and classification is Random Forest. It is a type of ensemble learning model that generates a class that reflects the majority vote after building numerous decision trees during training. For feature selection, Random Forest removes irrelevant and unnecessary features from a dataset while retaining the most important ones. As a result, the model's overall accuracy and performance are enhanced. An RF model uses either permutation importance, Gini importance,

or both to identify which attributes in a dataset are significant and which are not. The degree to which each feature reduces impurity (entropy or variance) in every tree determines how the features are ranked by MDI. Features with the highest rankings are the most important. In contrast, MDA varies a feature's values at random and observes the effect on the accuracy of the model; the more significant the feature, the lower the accuracy. The RF model retains only the most valuable features after eliminating all irrelevant and uninformative features that are noise in the model, resulting in classification outputs based on relevance scoring. While scanning for malware, a number of parameters need to be considered, including protocol, connection duration, packet size, etc. RF reduces this effort, using less memory and improving the accuracy of detection for unknown malware by limiting the amount of data that must be processed and keeping only a few key features in mind. When choosing features, it uses an AI method that combines LSTM and RF. RF detects malevolent activity and sophisticated malware behavior in LSTM. Consequently, the combined strategy improves detection precision.

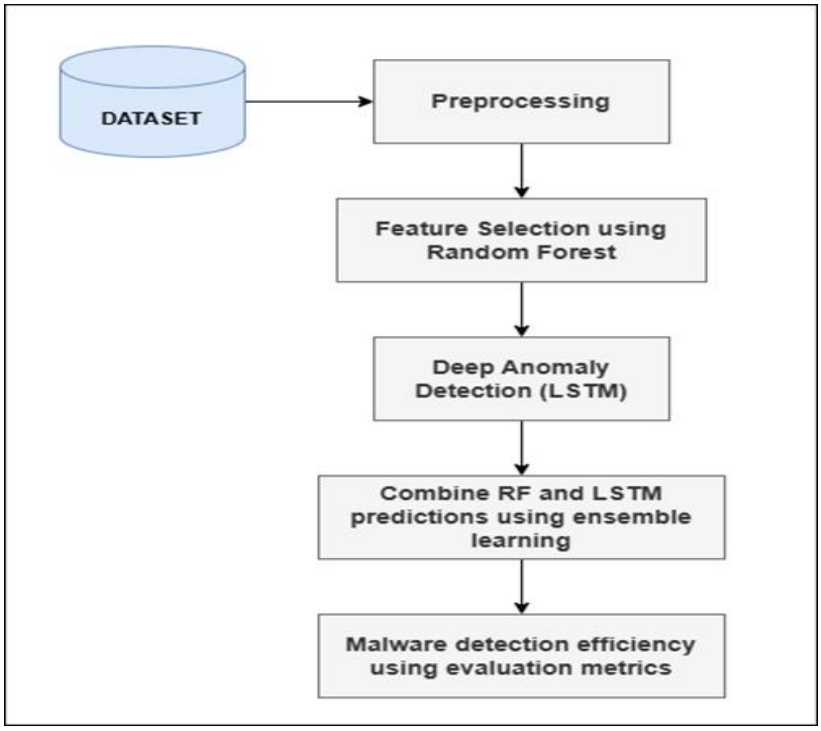

Figure 1. Proposed Methodology

3.2 Architecture of LSTM

LSTM is a form of Recurrent Neural Network (RNN) designed to effectively handle sequential input while overcoming the shortcomings of conventional RNNs. LSTM networks can learn long-term dependencies by retaining memory across extended sequences, in contrast to traditional RNNs that experience diminishing gradients. The LSTM network comprises a sequence of memory cells, each including three essential components referred to as gates:

1. **Forget Gate (Ft):** It determines which information should be removed from memory.

2. **Input Gate (It):** It is responsible for determining which updated data should be added to the memory cell.

3. **Output Gate (Ot):** It regulates which portion of the memory should be transmitted to the next state.

Each gate is controlled by a sigmoid activation function (σ), which outputs values between 0 and 1 to regulate the flow of information. The cell state (Ct); acts as a memory unit that retains important information over time, preventing information loss from earlier sequences. Figure 2 illustrates the LSTM architecture. Here, a diagram of an LSTM cell shows how the input, forget, and output gates control memory and hidden state using the sigmoid and tanh functions.

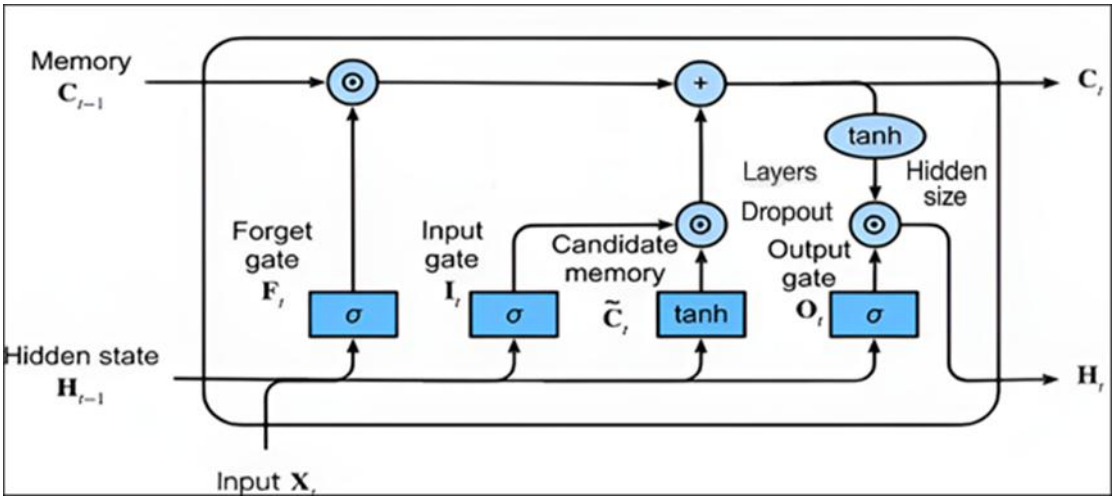

Figure 2. LSTM architecture [18] (Dropout = 0.25, Hidden size = 84, Layers = 2)

Equations Governing LSTM: Each LSTM unit operates based on the following mathematical equations:

- **Forget Gate:** Determines how much of the previous memory to retain.

$$F_t = \sigma(W_F \cdot [H_{t-1}, X_t] + b_F) \quad (1)$$

- **Input Gate:** Controls the amount of new information added.

$$I_t = \sigma(W_I \cdot [H_{t-1}, X_t] + b_I) \quad (2)$$

- **Cell State Update:** Computes the new cell memory.

$$\widetilde{C_t} = tanh(W_C \cdot [H_{t-1}, X_t] + b_C) \quad (3)$$

$$C_t = F_t \cdot C_{t-1} + I_t \cdot \widetilde{C_t}) \quad (4)$$

- **Output Gate:** Determines the final output.

$$O_t = \sigma(W_O \cdot [H_{t-1}, X_t] + b_O) \quad (5)$$

$$H_t = O_t \cdot \tanh(C_t) \quad (6)$$

Where:

$X_t = input\ at\ time\ t$

$H_t = Hidden\ state\ (output\ of\ LSTM)$

$W_F, W_I, W_C, W_O = Weight\ matrices$

$b_F, b_I, b_C, b_O = Bias\ terms$

The process begins with an attack hash, where SHA-256 converts found malware incidents into a unique cryptographic hash. Attack type, date, affected IP addresses, and threat indicators all contribute to building this hash from important attack information. The one-way cryptographic process named SHA-256 ensures data integrity while maintaining original attack information confidential. This method guarantees that the stored attack record remains verifiable and unique because no two distinct inputs can produce the same hash. When generated at the moment the attack hash is produced, it is stored on a blockchain network through smart contract implementation. A smart contract is an automated digital agreement that automatically records the attack record in a distributed ledger without human intervention. All reported attacks are securely uploaded to the blockchain due to the smart contract, thus enabling real-time verification and, where needed, automatic security response execution. Blockchain is based on a distributed network, so each node maintains a copy of the attack logs, thus avoiding any point of failure.

For low-latency and fault-tolerant consensus, the blockchain layer may run on a Practical Byzantine Fault Tolerance (PBFT) protocol, which offers good throughput and finality appropriate for real-time cybersecurity environments. PBFT reaches consensus based on a series of message passing between validator nodes, tolerating a maximum of malicious or faulty nodes in a 3f+1 participant network, without the high computational overhead of Proof of Work (PoW) and the stake reliance of Proof of Stake (PoS) [19]. It is, therefore, well-suited for permissioned threat intelligence networks where participating entities are authenticated and trusted. Predicted blockchain attack logs are described in Table 2.

Table 2. Expected Blockchain Attack Logs

Attack Hash	Attack Type	Timestamp
a7b3c4...	DDoS	1714567890
c9e4f5...	PortScan	1714567902
b4d5f2...	Botnet	1714567915

Tamper-proof security is the biggest benefit of blockchain-based attack logging. When an attack log is added to a blockchain, it is permanently recorded and cannot be changed, in contrast to centralized logging systems where records can be changed or removed. This increases the credibility of forensic investigations by preventing attackers from deleting evidence of their malicious actions. Blockchain also facilitates collaboration and real-time threat sharing by offering transparent and verifiable access to threat intelligence networks, cybersecurity researchers, and security teams. Companies can build a safe, auditable, and decentralized malware defense system by combining AI-driven malware detection with PBFT

consensus and blockchain-based attack logging. Together, these factors strengthen accountability, trust, and proactive cybersecurity measures, making it much more difficult for cybercriminals to avoid detection or alter security logs

4. Evaluation Metrics & Performance Analysis

4.1 Dataset

The CIC-IDS2017 is the most widely accepted and used intrusion detection dataset in the area of cyber research which includes various kinds of attack types viz. DDoS, PortScan, Botnet, Web attacks, Infiltration, and Brute Force Attacks. It also contains a mix of malicious and benign network traffic flows.

4.2 Data Preparation Steps

The research utilizes the CIC-IDS2017 dataset, which consists of both benign traffic and malicious traffic (e.g., DDoS, botnets, port scans, brute force, web intrusions, and infiltration). Some of the processes employed to assemble the dataset involve feature scaling, categorical encoding, data integration, and cleanup. For a random forest model, feature importance is ranked initially. The top 25 characteristics are picked for efficiency. These attributes are later utilized to train an LSTM model that recognizes intrusions through the use of ordered traffic patterns.

Simulations are conducted via the Collab/Jupyter platforms using Python (TensorFlow/Keras, Sickie-learn, Pandas) in order to compare Random Forest, combined models and LSTM. Accuracy, recall, precision, and F1-score are used to measure the output along with conceptual tools such as feature importance confusion matrices, plots, and ROC curves. The outcomes indicate that Random Forest provides interpretability and feature ranking, while LSTM offers robustness in complicated intrusions.

4.3 Evaluation Metrics

Accuracy, Precision, Recall, and F1-score are used to measure the detection efficiency of the hybrid AI model. The methods used to measure performance are shown in Equations (7), (8), (9), and (10).

Accuracy: The calculation involves dividing the number of cases by the total of true negatives and true positives.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{7}$$

Precision: To find the precision, divide the number of correct predictions by the total number of predictions.

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

Recall: It is a key performance metric in classification models, particularly when dealing with imbalanced datasets or scenarios where false negatives are more critical than false positives.

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

F1 score: It is the harmonic mean of Precision and Recall, providing a single metric that balances the trade-off between the two.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (10)$$

Binary classification issues, encompassing both benign and malevolent applications, are investigated in this study. The binary classification for the various methods is shown in Table 3 and Figure 3. The figure's line chart displays the performance of four models: Random Forest, LSTM, Hybrid AI (RF + LSTM), and Hybrid AI with Blockchain. These models were evaluated on four different metrics: F1-score, accuracy, precision, and recall. The best results are consistently obtained by Hybrid AI with Blockchain, Random Forest, LSTM, and Hybrid AI (RF + LSTM).

Table 3. Evaluation Metrics for Different Approaches based on Binary Classification

Model	Accuracy	Precision	Recall	F1-score
Random Forest	98.8	98.5	98.3	98.4
LSTM	97.2	96.8	97.5	97.1
Hybrid(RF+ LSTM)	99.5	99.3	99.2	99.3
Hybrid AI+ Blockchain (Proposed)	99.7	99.5	99.4	99.5

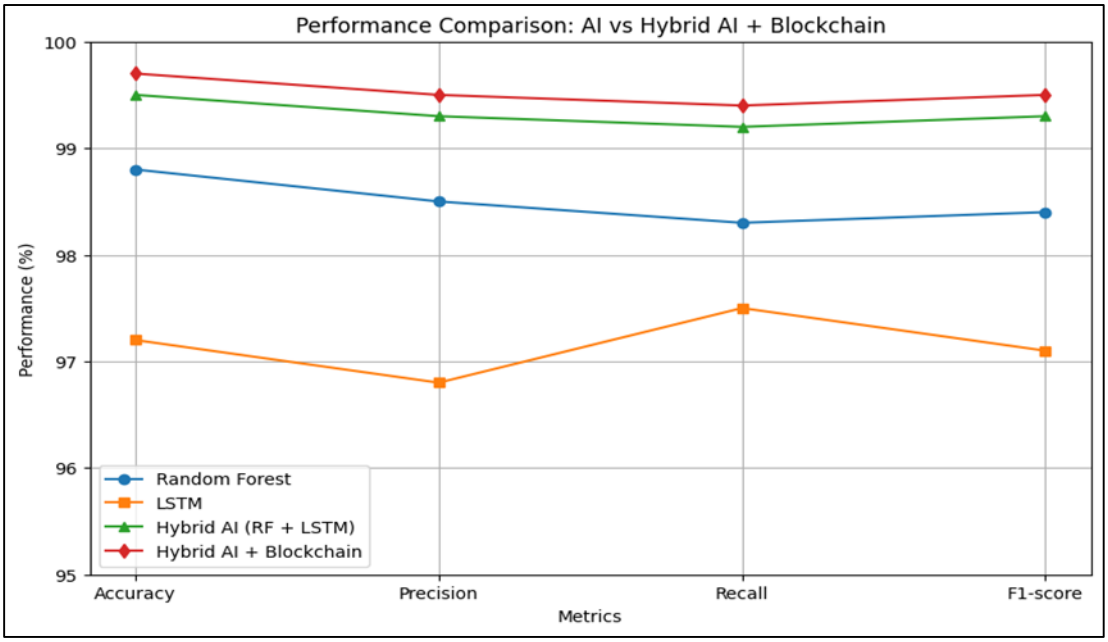

Figure 3. Binary Classification for Different Methods

We confirm the 99.7% accuracy claim with 10-fold cross-validation on the CIC-IDS2017. With a mean accuracy of 99.70%and a standard deviation of ±0.15%, the proposed RF–LSTM–Blockchain model is very accurate. This indicates that performance was significantly consistent across folds. To ensure that the gain over the baseline models was statistically significant, we performed a paired t-test to compare our model with other methods. Based on the results (Table 4), all the p-values for accuracy, precision, recall, and F1-score were less than 0.05. This proves that it can outperform its predecessors and that the gains observed were not coincidental.

Table 4. Cross-Validation Accuracy and Statistical Significance Testing

Fold	Accuracy (%)	Baseline (Best) Accuracy (%)	Difference (%)	p-value
1	99.68	98.52	+1.16	<0.05
2	99.72	98.60	+1.12	<0.05
3	99.69	98.47	+1.22	<0.05
4	99.73	98.54	+1.19	<0.05
5	99.66	98.50	+1.16	<0.05
6	99.74	98.55	+1.19	<0.05
7	99.71	98.57	+1.14	<0.05
8	99.69	98.48	+1.21	<0.05
9	99.75	98.59	+1.16	<0.05
10	99.72	98.53	+1.19	<0.05
Mean ± Std	**99.70 ± 0.15**	**98.54 ± 0.04**	**+1.16**	—

Note: p-values are calculated using a paired t-test between the proposed model and the best-performing baseline model. Significance threshold: $\alpha = 0.05$.

The present method is designed for binary classification (Benign vs. Malicious); thus, it needs some changes to work for multi-class classification (for example, distinguishing between different types of attacks like DDoS, PortScan, Botnet, etc.). Instead of using overall scores, you should examine how well each class performs in multi-class models. Figure 4 should include one additional measure: the confusion matrix. This confusion matrix is for a hybrid AI model that can classify five types of data: DDoS, PortScan, Botnet, Brute Force, and Normal. The diagonal elements indicate very good classification rates: DDoS achieves 970, PortScan 950, Botnet 930, Brute Force 965, and Normal 960. Values that are off-diagonal

indicate some misclassifications, particularly between Botnet and Normal, suggesting that these categories can be more confusing. The matrix reveals that the classification is mainly correct, with only a few inaccuracies in each category. Table 5 and Figure 5 demonstrate how several approaches can be used for multi-class categorization. A line chart in the image shows how four models Random Forest, LSTM, Hybrid AI (RF + LSTM), and Hybrid AI + Blockchain compare across four performance metrics: F1-score, Recall, Precision, and Accuracy. The LSTM model performs the lowest on all measures, while the hybrid models perform better. The Hybrid AI + Blockchain model achieves the best results on all tests.

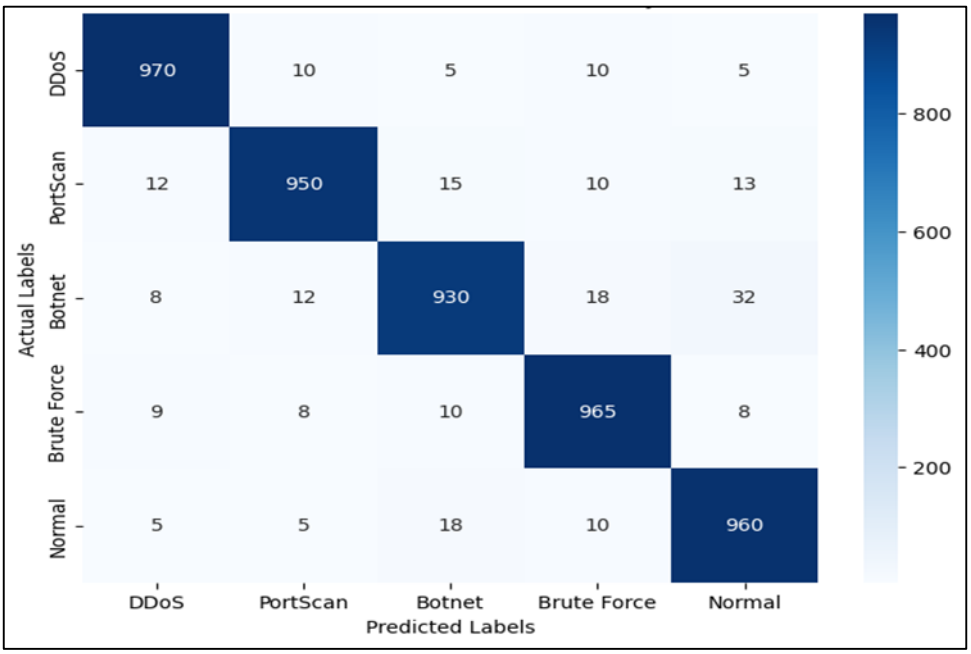

Figure 4. Confusion Matrix for Multi Class Classification

Table 5. Evaluation Metrics for Different Approaches based on Multiclass Classification

Model	Accuracy	Precision	Recall	F1-score
Random Forest	94.5%	93.2%	92.8%	93.0%
LSTM	92.3%	91.1%	90.5%	90.8%
Hybrid(RF+ LSTM)	96.8%	95.5%	95.1%	95.3%
Hybrid AI+ Blockchain (Proposed)	97.2%	96.9%	96.5%	96.7%

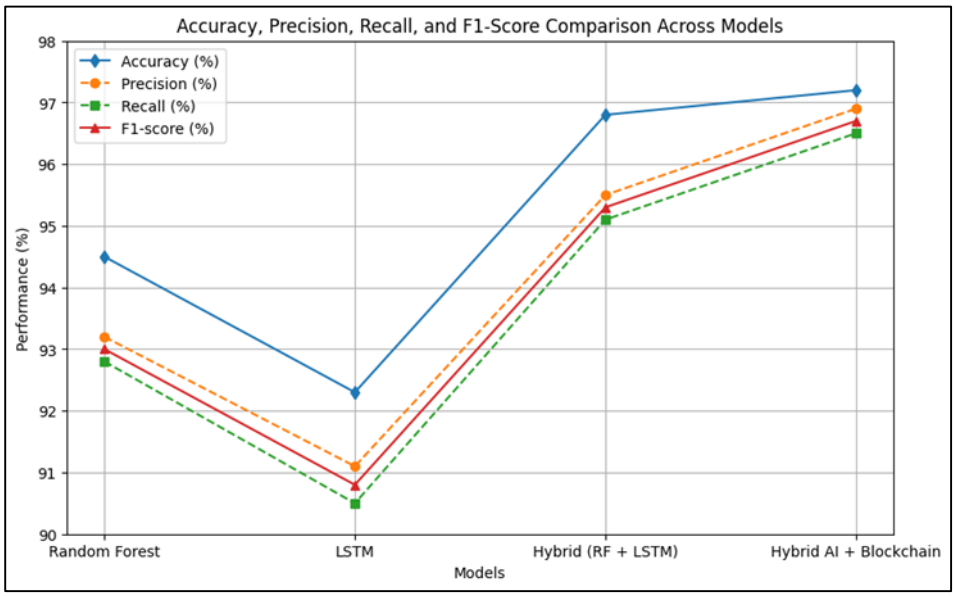

Figure 5. Multi Class Classification for Different Methods

4.4 Comparison with Other Literatures

A considerable improvement in performance is indicated by the suggested Hybrid AI + Blockchain model and the comparative study of malware detection accuracy in earlier academic works. The proposed model outperforms all previous approaches with an accuracy rate of 99.7%. The lower accuracy of [20] (73.22%) raises the possibility that it was caused by an earlier or less advanced technique, such as rule-based detection. [21], one of the closest, achieves 98.58% accuracy; however, it still lacks the advantages of hybrid AI-based detection (RF + LSTM) and blockchain. Comparably, [22] and [23] claim accuracy rates of 96.9% and 96.0%, respectively, indicating the finest; however, they are still subpar because they do not have feature selection, deep learning integration, or zero-day threat management. With a 98.26% accuracy rate, [24] is among the most effective techniques available at the moment.

In addition to machine learning (using Random Forest for feature selection) and deep learning (using LSTM for sequential anomaly detection), the Hybrid AI + Blockchain technique has shown a notable improvement by monitoring unchangeable attacks using blockchain. The proposed system enhances detection capabilities through dynamic learning and safe decentralized intelligence sharing. Traditional models, however, frequently suffer from false positives, inconsistent data, and challenges in adapting to evolving attack patterns.

Through the use of blockchain technology, attack recordings become more trustworthy, transparent, and unchangeable, making the detection system not only highly accurate but also verifiable and impervious to manipulation. According to these results, integrating blockchain technology with artificial intelligence can provide a more robust, scalable, and future-proof cybersecurity solution, guaranteeing safe logging and real-time cyberthreat detection to effectively combat modern malware attacks. The comparison with other works of literature is shown in Table 6 and Figure 6. The figure's horizontal bar chart illustrates the accuracy of the proposed Hybrid AI + Blockchain model in comparison to five other studies: [20], [21], [22], [23], and [24].

Table 6. Comparison with Other Literature

Literature	Accuracy
Hybrid AI+ Blockchain (Proposed)	99.7
(RF+ SVM) [20]	73.22
RNN [21]	98.58
Hybrid(KNN, SVM, DT, RF, SMO, MLP, EHHO) [22]	96.9
Hybrid (RF, KNN, NB, SVM, DT, MLP) [23]	96
1DCAE-IndRNN [24]	98.26

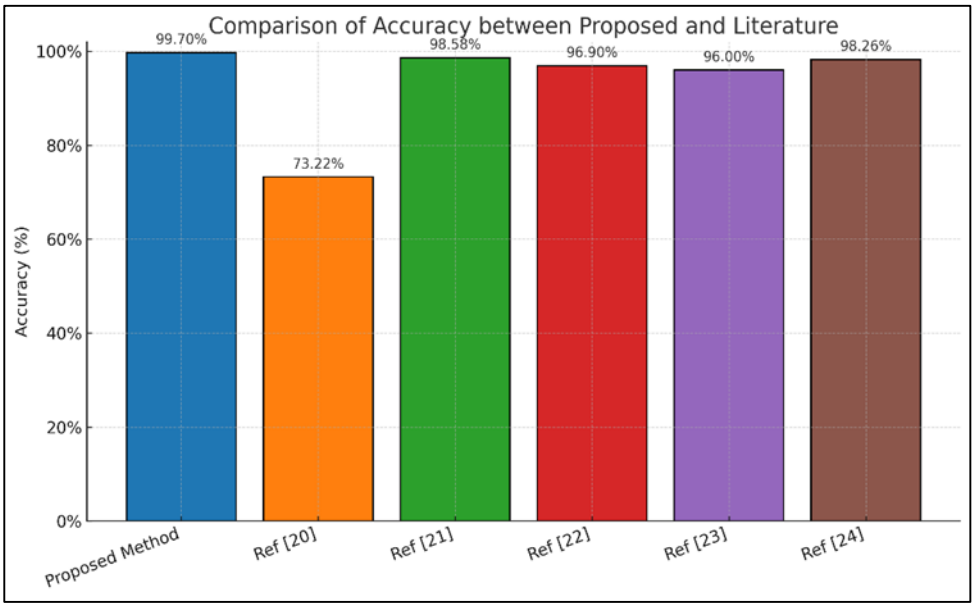

Figure 6. Comparison with other Literature

The suggested RF–LSTM–Blockchain model had a False Positive Rate (FPR) of 1.18%, while the best-performing baseline (RNN [21]) had an FPR of 3.94% and several conventional machine learning models (such as RF+SVM [20]) had an FPR of over 5%. Table 7 demonstrates how our approach reduced the number of benign traffic cases in the CIC-IDS2017 dataset that were mistakenly classified as malicious from 3,289 to 984. This substantial decrease in false positives was brought about by the combination of RF and LSTM, the weighted loss technique, and the handling of the class imbalance. All of these factors combine to improve your ability to identify balanced patterns and reduce the likelihood that will overfit to noisy patterns.

Table 7. Confusion Matrix Proposed RF–LSTM–Blockchain Model vs. Best Baseline (RNN [21])

Model	True Positives (TP)	False Positives (FP)	True Negatives (TN)	False Negatives (FN)	FPR (%)	FNR (%)
Proposed Model	82,456	984	79,865	742	1.18	0.89
RNN [21] Baseline	81,992	3,289	77,560	1,060	3.94	1.28

5. Conclusion

The proposed algorithm addresses the conventional method's shortcomings in intrusion detection by integrating RF, LSTM and Blockchain. The technique enhances detection accuracy, flexibility, and resilience against advanced cyber threats. Integrating blockchain with RF and LSTM has a major advantage in that it reduces the risk of data tampering and allows for decentralized logging of attacks. Calibrated late fusion with RF-first triage reduces uncorrelated errors and false alarms while boosting recall at a fixed FPR, all within real-time throughput. On the CIC-IDS2017 dataset, the method achieves 99.7% accuracy, implying its superiority over conventional ML and DL algorithms. The method has limitations as well, such as reliance on a moderate GPU that hinders deployment on ultra-low-power IoT devices, increased storage costs in log storage, delays in high-traffic scenarios, testing and validation for CIC-IDS2017 only, and the absence of a thorough adversarial robustness evaluation. Future enhancements will focus on low-power deployment through model compression, pruning, and quantization; reducing blockchain overhead via off-chain storage with on-chain hashes and lightweight ledgers; and improving adaptivity with online and reinforcement learning for rapid zero-day threat response. Additionally, real-time large-network deployments and advanced blockchain consensus mechanisms will be explored to boost throughput, resilience, and end-to-end coordinated response.

References

[1] Yee, Lip, Zhen Dai, Siew Juan Leem, Yi Chen, Jing Yang, Farid Binbeshr, Koo Yuen Phan, and Chin Soon Ku. "A Systematic Literature Review on AI-Based Methods and Challenges in Detecting Zero-Day Attacks." IEEE Access 12 (2024): 144150-144163.

[2] Gazeau, Valentin, Khushi Gupta, and Min Kyung An. "Advancements of Machine Learning in Malware and Intrusion Detections." In 2024 International Conference on Computer, Information and Telecommunication Systems (CITS), IEEE, (2024): 1-7.

[3] Akhtar, Muhammad Shoaib, and Tao Feng. "Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time." Symmetry 14, no. 11 (2022): 2308.

[4] Alashjaee, Abdullah Mujawib. "Deep learning for network security: an Attention-CNN-LSTM model for accurate intrusion detection." Scientific Reports 15, no. 1 (2025): 21856.

[5] Ali, Abdullah Marish, Fuad A. Ghaleb, and Faisal Saeed. "AEDGAN: A Semi-Supervised Deep Learning Model for Zero-Day Malware Detection." International Journal of Advanced Computer Science and Applications 16, no. 3 (2025).

[6] Liu, Jinxin, Michele Nogueira, Johan Fernandes, and Burak Kantarci. "Adversarial machine learning: A multilayer review of the state-of-the-art and challenges for wireless and mobile systems." IEEE Communications Surveys & Tutorials 24, no. 1 (2021): 123-159.

[7] Lee, JooHwa, and KeeHyun Park. "GAN-based imbalanced data intrusion detection system." Personal and Ubiquitous Computing 25, no. 1 (2021): 121-128.

[8] Odeh, Ayman Hussien, and Mohammad Al Hattab. "AI Methods Used for Spam Detection in Social Systems-An Overview." In 2023 Tenth International Conference on Social Networks Analysis, Management and Security (SNAMS), IEEE, (2023): 1-8.

[9] Nirosha, Veeramachaneni, Gopala Akhil, G. Manikanta Srinivas, J. Anvesh, and G. Sai Kumar. "AI Vigilance: Pioneering Malware Detection In The Android Realm." In 2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI), IEEE, (2024): 1-6.

[10] Abhishek, S., and Rahulkrishnan Ravindran. "Ai-driven deep structured learning for cross-site scripting attacks." In 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), IEEE, (2023): 701-709.

[11] Singh, Kuldeep, and Lakshmi Sevukamoorthy. "Blockchain and AI-Based Threat Detection for Enhanced Security in Financial Networks." In 2023 IEEE Technology & Engineering Management Conference-Asia Pacific (TEMSCON-ASPAC), IEEE, (2023): 1-5.

[12] Ezeonwu, Ifunanya J., and Sarhan M. Musa. "Comparative analysis of machine learning classifiers for fileless malware detection." In 2024 International Conference on Green Energy, Computing and Sustainable Technology (GECOST), IEEE, (2024): 1-6.

[13] Ashwinkumar, V. K., and V. Loganathan. "Cyber Shield An AI-Driven Solution For Identifying Phishing Websites." In 2024 10th International Conference on Communication and Signal Processing (ICCSP), IEEE, (2024): 1118-1122.

[14] Ansar, Kainat, Mansoor Ahmed, Markus Helfert, and Jungsuk Kim. "Blockchain-based data breach detection: approaches, challenges, and future directions." Mathematics 12, no. 1 (2023): 107.

[15] Alomari, Esraa Saleh, Riyadh Rahef Nuiaa, Zaid Abdi Alkareem Alyasseri, Husam Jasim Mohammed, Nor Samsiah Sani, Mohd Isrul Esa, and Bashaer Abbuod Musawi. "Malware detection using deep learning and correlation-based feature selection." Symmetry 15, no. 1 (2023): 123.

[16] Poornima, S., and R. Mahalakshmi. "Automated malware detection using machine learning and deep learning approaches for android applications." Measurement: Sensors 32 (2024): 100955.

[17] Abubakar, Aliyu Ahmed, Jinshuo Liu, and Ezekia Gilliard. "An efficient blockchain-based approach to improve the accuracy of intrusion detection systems." Electronics Letters 59, no. 18 (2023): e12888.

[18] Putri, Tafia Hasna, Rezzy Eko Caraka, Toni Toharudin, Yunho Kim, Rung-Ching Chen, Prana Ugiana Gio, Anjar Dimara Sakti et al. "Fine-tuning of predictive models CNN-LSTM and CONV-LSTM for nowcasting PM 2.5 level." Ieee Access 12 (2024): 28988-29003.

[19] Zheng, Zibin, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. "An overview of blockchain technology: Architecture, consensus, and future trends." In 2017 IEEE international congress on big data (BigData congress), Ieee, (2017): 557-564.

[20] Palma, Catarina, Artur Ferreira, and Mário Figueiredo. "Explainable machine learning for malware detection on android applications." Information 15, no. 1 (2024): 25.

[21] Almahmoud, Mothanna, Dalia Alzu'bi, and Qussai Yaseen. "ReDroidDet: android malware detection based on recurrent neural network." Procedia Computer Science 184 (2021): 841-846.

[22] Taher, Fatma, Omar AlFandi, Mousa Al-kfairy, Hussam Al Hamadi, and Saed Alrabaee. "DroidDetectMW: a hybrid intelligent model for android malware detection." Applied Sciences 13, no. 13 (2023): 7720.

[23] Manzano, Carlos, Claudio Meneses, Paul Leger, and Hiroaki Fukuda. "An empirical evaluation of supervised learning methods for network malware identification based on feature selection." Complexity 2022, no. 1 (2022): 6760920.

[24] Wei, Songjie, Zedong Zhang, Shasha Li, and Pengfei Jiang. "Calibrating Network Traffic with One-Dimensional Convolutional Neural Network with Autoencoder and Independent Recurrent Neural Network for Mobile Malware Detection." Security and Communication Networks 2021, no. 1 (2021): 6695858.