*Journal of Discrete Mathematical Sciences & Cryptography*
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 28 (2025), No. 4-B, pp. 1437–1449
DOI : 10.47974/JDMSC-2289

# A hybrid cryptographic and deep learning approach for IoT device security

Ahmed Fakhir Mutar *
Leyli Mohammad Khanli †
*Department of Computer Engineering*
*Faculty of Electrical and Computer Engineering*
*University of Tabriz*
*Tabriz*
*Iran*

Hojjat Emami §
*Department of Computer Engineering*
*Faculty of Engineering*
*University of Bonab*
*Bonab*
*Iran*

**Abstract**

A seamless network of connected things and automated processes have transformed a variety of industries thanks to the Internet of Things (IoT). In addition to introducing significant security vulnerabilities, IoT devices are also susceptible to cyberattacks due to their limited computing power. There is no doubt that traditional cryptographic techniques provide a foundation of security, but they are often inadequate to counter sophisticated attacks. A hybrid security framework is presented in this paper that combines enhanced Elliptic Curve Cryptography (ECC) algorithms with deep learning-based intrusion detection algorithms to address these challenges. Using an enhanced Elliptic Curve Cryptography (ECC) algorithm along with deep learning-based intrusion detection, this paper addresses these challenges. According to the performance evaluations, the proposed approach reduces encryption and decryption times, lowers memory consumption, and enhances processing

* *E-mail:* **diazrodr.od@gmail.com (Corresponding Author)**
† *E-mail:* **nguye365@msu.edu**
§ *E-mail:* **emami@ubonab.ac.ir**

© Taru Publications

efficiency. This hybrid cryptographic/deep learning approach improves IoT security, providing a scalable and reliable means of protecting IoT networks against emerging threats.

***Subject Classification:*** *93E10, 94A13.*

***Keywords:*** *IoT security, Elliptic curve cryptography (ECC), Deep learning-based intrusion detection, Anomaly detection, Hybrid cryptographic framework.*

## 1. Introduction

Internet of Things (IoT) has revolutionized many industries, enabling seamless connectivity and automation across a wide range of fields. These expansions have also presented significant security challenges due to IoT devices' limited computational resources and weak security mechanisms. IoT networks can be secured using traditional cryptographic techniques, but sophisticated cyber-attacks may exceed their capabilities. IoT device security can be enhanced by integrating cryptographic security methods with deep learning techniques based on this paper's hybrid approach. Data confidentiality, integrity, and real-time threat detection are ensured through the combination of encryption algorithms and AI-driven anomaly detection. In a hybrid model, lightweight cryptographic protocols are leveraged to ensure secure communication. In contrast, deep learning models are employed to identify and mitigate potential security threats on a case-by-case basis. As the Internet of Things (IoT) has gained traction in recent years, countless things have been connected across industries, households, and transportation [1]. Microcontrollers enable IOT-enabled devices to monitor temperatures in a variety of household appliances, including refrigerators, microwaves, washing machines, air conditioners, and televisions [2]. Using the Internet, these gadgets send data to the cloud server, which communicates with it via the Internet. Because wireless communications are highly secure, signals are broadcast across mediums, allowing unauthorized users to access them [3]. Since IoT platforms rely heavily on wireless communication and devices are entering the IoT environment at an increasing rate, security has become increasingly important [4]. A cyberattacks can steal the most personal information from internet-connected devices, which are most vulnerable to cyber-attacks [5]. A growing number of connected devices and a growing amount of data generated by those devices pose the biggest threat to IoT security.

IoT devices connect to secure servers using TLS and verify the certificate of the secure server during a handshake. With the dynamic key generation scheme, IoT devices are more secure during data transmission. Improved performance and security of wireless networks can be achieved by using key generation. An anomaly intrusion detection system (IDS) also uses typical behaviour criteria to identify actions that differ significantly from standard behaviour. A node's characteristics determine what type of malware attack it is in the Internet of Things network. This study proposed a deep LSTM and

improved ECC algorithm for detecting malware in IoT devices. An IoT network aggregates, processes, and alters data from each interconnected device for the purpose of providing additional services [6]. Cyberattacks can be thwarted using various cryptographic tools, as illustrated in Figure 1.

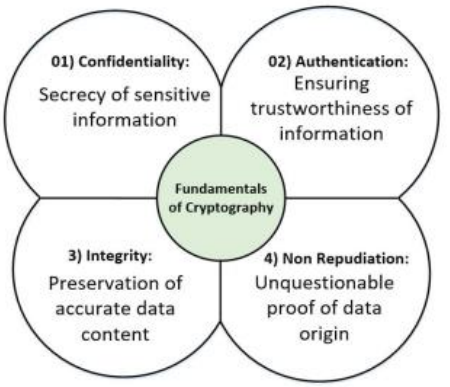


**Figure 1**
**The key elements of cryptographic protection against cyberattacks.**

APTs (Advanced Persistent Threats), (Ransomware), and Phishing can be used to exploit and steal sensitive information [7]. Businesses and individuals suffer significant financial losses as well as reputational damage and distrust from these incidents. The impact of cyber threats on critical infrastructure can go beyond financial impacts. The best way to prevent data theft is to employ robust approaches like encryption protocols, segmented networks, and access controls that incorporate intrusion detection systems [8]. Collaboration, policies, and education are also important components of cybersecurity. A collaborative government-industry-government approach is necessary for cybersecurity education. In this way, attacks can be better understood, and digital data can be kept more secure and confidential.

## 2. Related Work

IoT devices are susceptible to cyber threats, which have led to a growing concern about their security. IoT security can be enhanced in a number of ways, including machine learning and cryptography. A wide range of industries have been revolutionized by the Internet of Things (IoT), including healthcare, which has seen remote patient monitoring and smart implants become increasingly popular in recent years [9], smart homes, where appliances and sensors are interconnected [10], as well as industrial settings, where the Industrial Internet of Things (IIoT) connects appliances and sensors [11]. As technology is more and more integrated into our daily lives, the attack surface has grown significantly. There is no doubt that the number of interconnected devices, the often-limited processing capabilities, and the wide range of communication

protocols employed create a complex and vulnerable ecosystem. As this expanding landscape presents multifaceted challenges, traditional security measures relying primarily on cryptography [12] are proving inadequate. Cyber threats are becoming sophisticated and evolving, necessitating a more robust and adaptive security approach to address these challenges [13].

ML can be used in multiple fields to improve services, according to the author [14]. Additionally, IoT plays a major role in facilitating human interaction on one device. Furthermore, the advent of digital technology has made it possible to transmit data without human interaction. Security and privacy patterns were addressed through the use of machine learning. A key objective of the study was to improve the effectiveness of recognizing attacks by applying machine learning (ML). Considering the security and privacy concerns associated with the Internet of Things was a major focus of the author's research [15]. Several IoT networks have been compromised, and human security is at risk due to IoT attacks. This paper proposes to detect attacks through reinforcement learning and convert them into targeted attacks. As part of this work, we explored the features of IoT traffic and used entropy-based metrics to forecast attacks in IoT. In addition, the developed model was proven to be effective using the IoT dataset. IoT networks use deep learning to detect anomalies and predict threats. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are AI-driven models that can successfully detect malicious activity. The models, however, require significant computation resources, and adversaries can attack them. Researchers recently found that cryptographic techniques and deep learning techniques can be used to strengthen IoT security frameworks. Data confidentiality and real-time threat detection can be enhanced by combining encryption with AI-based threat detection, according to a study by [16].

## 3. Proposed Methodology

### 3.1 *Pre-processing using Normalization*

There may be missing data and repetitive packets in the input data because it has not been processed. In addition to eliminating duplicates and redundant occurrences, missing data has been tidied up. It is necessary to employ sample size minimization techniques to deal with the extensive dataset for the education system. To exclude the non-significant attributes from this dataset, feature extraction tools are needed. As part of the pre-processing stage, it is possible to normalize the dataset. The first step in normalizing is to generate a z-score, which can be expressed via Equation (1).

$$Z = \left[\frac{(Y-\alpha)}{w}\right] \tag{1}$$

A dataset α represents a mean and ω standard deviation, respectively, with Z given by Equation (2).

$$Z = \frac{Y-\bar{Y}}{D} \tag{2}$$

A sample mean is denoted $\bar{Y}$, and a standard deviation is denoted D.

According to Equation (3), a random sample must follow the same pattern.

$$Z_j = \beta_0 + \beta_1 \gamma_j + \varepsilon_j \tag{3}$$

In the following example, $\varepsilon_j$ denotes the errors. $w^2$ relies on the errors, so the errors should not depend on one another.

$$y_j \sim \sqrt{w} \frac{y}{\sqrt{y^2+w-1}} \tag{4}$$

$Y$, a random variable is represented by this expression, which is used to compute its moment scale deviation. Based on the standard deviation of the variables' movements, the following expression is used to calculate the moment scale deviation.

$$M = \frac{\lambda^m}{\phi^m} \tag{5}$$

A moment scale is represented by $m$ here.

$$\lambda^m = E(Y-\alpha)^M \tag{6}$$

A random variable is represented by $Y$, and an expected value by $E$.

$$\phi^m = \left(\sqrt{E(Y-\alpha)^M}\right)^2 \tag{7}$$

$$y^w = \frac{m}{Y} \tag{8}$$

A variance coefficient is represented by $y^w$.

When all values are set to 0 or 1, the feature scaling procedure is stopped. This procedure is called unison-based normalizing. In the normalized Equation, we will write,

$$Y' = \frac{(y-y_{min})}{(y_{max}-y_{min})} \tag{9}$$

Data sets can be managed once they have been normalized so that their extent and variability remain constant. Latency reduction or elimination is the goal of this stage. This will result in normalized data being fed into the following phases:

### 3.2. *Feature Selection and Extraction*

Using PSO, one can simulate the behaviour of birds and fish through computational optimization. PSO is a method for finding the optimal solution to a problem by navigating potential solutions to it (particles) through a search space. It is common for researchers to simplify the PSO algorithm as a random search challenge in a multi-dimensional space (D-dimensional space). An objective function must be optimized as the primary objective. A population in a D-dimensional space consists of n particles. $P_k = (P_{k1}, P_{k2}, \ldots., P_{kD})$, each of which has a position vector $x_k = (x_{k1}, x_{k2}, \ldots., x_{kD})$ in the D-dimensional space. In a population, a fitness function measures each particle's fitness. By introducing a hyperparameter $\alpha$, the relationship between classifier performance $Q$ and feature subset $N_g$ in relation to the total number of features $N_u$ can be managed

$$F(X) = \alpha(1-Q) + (1+\alpha)\left(1 - \frac{N_g}{N_u}\right) \tag{10}$$

By iteratively exploring D-dimensional space, particle k begins with a set of randomly positioned particles and gradually converges on an optimal solution. This self-discovered optimal position $P_k = (P_{k1}, P_{k2}, \dots., P_{kD})T$ serves as a local optimal solution, which is characterized by its velocity vector $v_k = (v_{k1}, v_{k2}, \dots., v_{kD})T$. An optimal position $P_g = (P_{g1}, P_{g2}, \dots., P_{gD})T$ represents the global optimal solution, which is determined by a collective search within a particle swarm. Particles adjust their position and velocity throughout each iteration by tracking two optimal solutions, namely $(P_i,\ P_g)$. According to Equations (12) and (13), there is a formula for updating the system.

$$v_{kd}(T+1) = wv_{kd}(T) + clr1(P_{kd}(T) - x_{kd}(T) + c2r2\left(P_{gd}(T) - x_{kd}(T)\right) \tag{11}$$

$$k = 1,2, \dots, N: d = 1,2, \dots, D \tag{12}$$

A particle $k$ has a d-th dimension and is denoted by $N$, the total number of particles in the population. Iteration number $T$ refers to the current iteration, and $\omega$ refers to a non-negative inertia factor that governs the balance between local and global optimization. As $\omega$ increases, global optimization becomes stronger while local optimization becomes weaker, and vice versa.

### 3.3. *Anomaly Detection Using Deep Learning*

Most commonly used in deep learning, it is better suited for sequence forecasting problems. Data can be processed without images like single data points as long as it consists of a sequence of data. Many issues are resolved more quickly with it. As the data from the cell state is added and reduced, the LSTM's main function is to perform basic computations. As support for this mechanism, sigmoid neural networks and point-wise multiplication are used. LSTMs can also be enhanced with Bidirectional LSTMs. Recurrent nets can be segregated both forward and backward using this system. There is only one output layer connecting both sequences. In Bidirectional LSTM, each point in the provided sequence is fully described. An update gate is used to handle the hidden layer quantity, and this state is transmitted to the next layer.

$$X_v = \sigma(Z_x.[d_{v-1}, Y_v] + a_x) \tag{13}$$

$$e_v = \sigma(Z_x.[d_{v-1}, Y_v] + a_e) \tag{14}$$

$$d'_v = \tanh(Z.[e_v * ([d_{v-1}, Y_v] + a_d) \tag{15}$$

$$d_t = (1 - x_v) * d_{v-1} + x_v * d'_v \tag{16}$$

$$p_v = \sigma(Z_0.d_v) \tag{17}$$

Weight parameters for the input data are represented by $Z, Z_x$, and $Z_e$ reset gates are denoted by $X_v$ update gates are denoted by $y_v$, inputs of the current layer are denoted by $y_v$, outputs of the prior layers are represented by $d_{v-1}, \sigma$ specified as the sigmoid function, $a_d, a_e$, and $a_x$ are used to indicate biases. Using tanh, the network value is fixed. In this step, the final output is acquired, and the loss value is calculated using the loss function.

$$L_v = \frac{1}{2}(p_h - p_v^0)^2 \tag{18}$$

$$L = \sum_{v-1}^{l} L_v \tag{19}$$

3.3.2. LSTM-GRU model

A dataset of regressions and time series issues is used to create results, which are created using deep learning and classical methods. The LSTM-GRU model uses LSTM as an initial step, and in backpropagation, it is used to solve the vanishing gradient problem. An LSTM consists of three gates: an input gate ($r_t$), a forget gate ($u_t$), and an output gate ($l_t$). These gates are used to store data in memory. Using analogy, data are stored with gates that are multiplied component-by-component by the sigmoid function. The gate can reject data or leave it in memory if it is set to zero. As a function of minus 1 to plus 1, Tans is the best activating function. Data fading is prevented by utilizing a second derivative. Nonlinear activation functions are best represented by the sigmoid function, which has a range from 0 to 1. It decides whether to drop or store data. Mathematical equations are explained in Equations (20) to (22) by considering input gates ($r_t$), forget gates ($u_t$), and output gates ($l_t$). GRU has two gates, the reset gate ($f_t$) and the update gate ($i_t$). This method supplies the LSTM output to the GRU. There are three gates in LSTM: $c_r^g \sim X_g$ and $X_g$, and $r_t, u_t$, and $l_t$, each representing a feature space at a particular time and area. GRU has two gates, $f_t$ and $i_t$. LSTM-GRU has two hidden layers, $\sim X_g, \sim s_g$, and $s_g$. In LSTM, $d_r;\ d_u;\ d_l;$ and $d_x$ represent the weights. The GRU is weighted by $d_f\ ;\ d_i;\ d_l;$ and $d_{x_g}$, and the LSTM-GRU model is biased by $y_r;\ y_u;\ y_l;$ and $y_x$. Hyperbolic tangent function is tans, where v and t are the scalar products of two vectors. As $cg$ is transferred to the input network unit, its weight is multiplied by its bias ($y_r$) and the bias is added after $cg$ has been multiplied by its weight.

A sigmoid function transforms it into values between (0, 1), followed by updating the level of the cell.

$$r_t = \beta\left(d_r\left[s_{g-1}, c_g\right] + y_r\right) \tag{20}$$

$$u_t = \beta\left(d_u\left[s_{g-1}, c_r\right] + y_u\right) \tag{21}$$

$$l_t = \beta\left(d_l\left[s_{g-1}, c_r\right] + y_l\right) \tag{22}$$

Based on Equations (23) and (24) using the sigmoid function, we can generate results between 0 and 1. It is up to $\sim X_g$ and $X_g$ to decide what data will be stored in memory and what will be ignored. An $\sim X_g$ with a significant value calculated by multiplying it by a tan s function

$$\sim X_g = \tan S\left(d_x\left[s_{g-1}, c_g\right] + y_x\right) \tag{23}$$

$$X_g = u_g * X_{g-1} + r_g * \sim X_g, \tag{24}$$

This process is explained in Equation (25) and (26) by dividing ($s_{g-1}$) by ($f_t$), and multiplying this value by (weight), then passing the data to the reset gate ($i_t$).

$$f_t = \beta(d_f\left[X_g\right] + d_f\left[s_{g-1}\right] \tag{25}$$

$$i_t = \beta(d_i\left[X_g\right] + d_i\left[s_{g-1}\right] \tag{26}$$

$S_g$ decides what data to store. In addition to storing data, the output layer also contains the tan s, which is used to forecast traffic at a particular time and place. According to Equation (27) to (29), the above process takes place.

$$\sim s_g = \tan s(d_{x_g} + i_t^{\circ} dx_g[s_g - 1] \quad (27)$$

$$s_g = f_t^{\circ} s_{g-1} + (1 - f_t)^{\circ} \sim s_g \quad (28)$$

$$s_g = l_t * \tan s\,(s^g) \quad (29)$$

As part of the LSTM-GRU framework, four hidden layers each consist of 256 hidden units. A hidden layer is activated as a function of all other hidden layers. Dense layers use only one unit. Our test and training datasets are separated using holdout cross-validation. This is followed by 512 batches of training on the training dataset. A subsequent step is to determine whether the dataset's model can be generalized. Similar to the output layer's linear activation function, the hidden layer's tan function is the most effective.

### 3.4. *Enhanced Elliptic Curve Cryptography Algorithm (EECCA)*

An elliptic curve over a finite region is used as the basis for the EECCA, a public-key cyber security technique. A curve with an elliptic shape can be defined by Equation (30).

$$r^2 + qr = q^3 + cq + d^4 \quad (30)$$

Procedures for building elliptic curves: elliptic curves have the main advantage of making a third point by adding two locations on the curve. Design engineers may define multiplier $jQ$ as the total of $j$ duplicates of $Q$, where $j$ and $Q$ are positive numbers, and $Q$ is a point, respectively.

Encryption ensures the security of confidential information. The EECCA algorithm provides protection. Prime numbers and universal points are used to generate the private key. Prime numbers affect the creation of private keys. Compared to the existing ECC, the proposed algorithm further reduces key size, thereby minimizing computation overhead. Following are the steps for demonstrating the EECCA algorithm:

***Encryption Algorithm:*** Decide all the $Ep(x, y)$ points on a relevant curve$Ep(x, y)$. Throughout the curve, each point represents an alphabetical, numerical, or unique character. It is recommended to select the global point *P* with the higher-order m in $Ep(x, y)$. Private keys can be chosen by either the sender or receiver. Senders and receivers' public keys should be evaluated. Encryption is conducted using private data points. Estimating encrypted points is possible using the universal point P, a random integer, and the public key of the receiver. Index values are fed as inputs for assessment.

***Decryption Algorithm:*** It is necessary to feed index values generated by the encryption algorithm into the decryption process. The receiver's curve type must be $Ep(x, y)$, and the sender's index table must be similar. Taking the secret key of the receiver, an arbitrary number, and the universal point into account will lead to the encrypted points being decrypted. This data is plaintext, which is a table of equivalent characters for EECCA points.

## 4. Result and Discussion

For data transmission security in IoT networks, a comparison of different encryption algorithms is presented in Figure 2a. Analyses of existing methods, such as ECC, AES, and DES, are conducted to compare the proposed approach with them. According to the proposed method, encryption takes 6.4 seconds, while ECC, AES, and DES take 6.9 seconds, 8.2 seconds, and 9.5 seconds, respectively. The proposed algorithm achieves a lower encryption time compared to existing techniques. A comparison of the Internet of Things encryption algorithm is shown in the Figure 2b. The proposed approach, ECC, AES, and DES are compared with existing techniques. The proposed model decrypts in 0.0150 seconds, while ECC, AES, and DES take 0.0168, 0.0152, 0.0170, and 0.0178 seconds, respectively. When compared to existing algorithms, the proposed algorithm achieves a faster decryption rate

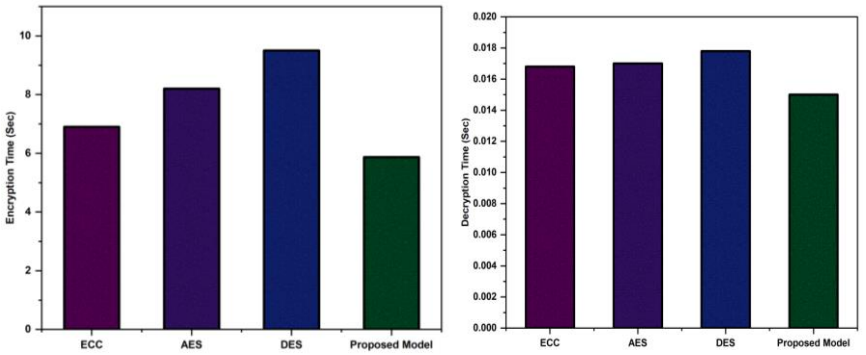


**Figure 2**

**Performance analysis of (a) encryption time (sec) and (b) decryption time (sec) versus number of algorithms.**

In Figure 3a, different encryption algorithms used in secure data transmission within IoT networks are compared based on their processing times. A comparison is made between IECC and existing encryption methods, such as ECC, AES, and DES. Compared to ECC, AES, and DES, which have processing times of 7 seconds, 8.2 seconds, and 9.6 seconds, respectively, the proposed model takes 6.2 seconds to process. Based on the results, the proposed algorithm is faster than existing algorithms. Based on these findings, the proposed model was demonstrated to be effective in enhancing security because of its superior performance.

As shown in the Figure 3b, various encryption algorithms used on the Internet of Things use different amounts of memory. According to the proposed approach, 51.22 MB of memory is consumed, while ECC consumes 75 MB, 79 MB is consumed by AES, and 81 MB is consumed by DES. As a result, the proposed approach requires significantly less memory than existing techniques, thus making it a more efficient method for transmitting secure data. According to Figure 4a, different file sizes result in different encryption times. Using the

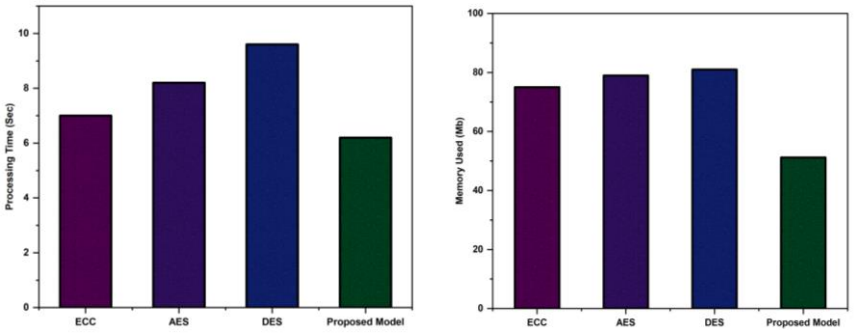


**Figure 3**
**Performance analysis of (a) processing time (sec) and (b) memory used (Mb) versus number of algorithms**

proposed approach, you can encrypt a file that is 2 MB, 4 MB, or 6 MB in 7 seconds, 13 seconds, and 19 seconds, respectively. This technique has been shown to be more efficient in terms of secure data transmission than existing techniques. Based on the proposed approach, the decryption time is 0.015 seconds with respect to file, respectively. It has been demonstrated that the proposed approach is more efficient in securing data transmission than existing techniques.

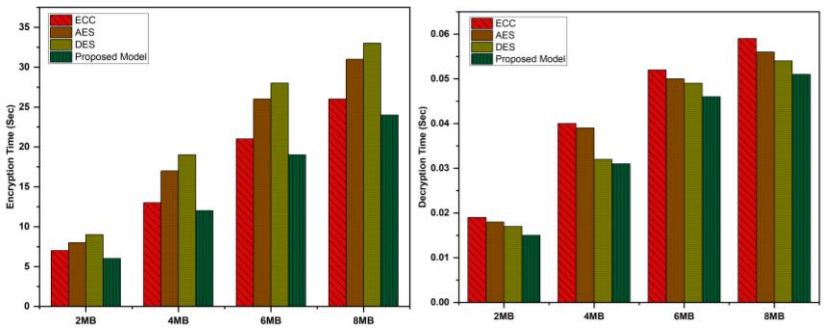


**Figure 4**
**Performance analysis of (a) encryption time (sec) and (b) decryption time (sec) versus file size.**

The proposed approach is validated by comparing it with existing approaches, including RNN [17] and DNN [18]. Comparing the proposed model with existing RNNs and DNNs, Figure 5 shows that it performs better.

## 5. Discussion and Conclusion

The security of IoT devices is enhanced by combining ECC encryption and LSTM anomaly detection by using a hybrid security framework. Secure data transmission and dynamic detection of malicious activity mitigate security risks

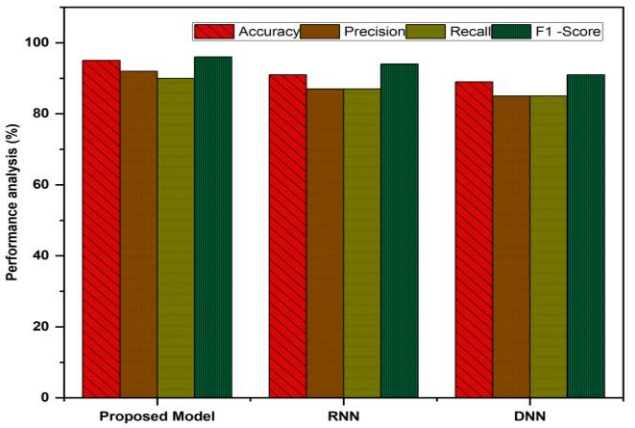


**Figure 5**
**Performance analysis versus existing algorithms**

effectively. According to the performance analysis, the proposed approach reduces encryption and decryption time, reduces processing overhead, and optimizes memory usage over conventional encryption. Deep learning is employed in this framework to detect anomalies, allowing it to be scalable and efficient when securing IoT devices. Future research may focus on enhancing computational efficiency and integrating quantum-resistant cryptographic techniques to enhance resilience against emerging attack vectors and enhance computational efficiency.

## References

[1] P. Rani and R. Sharma, "Intelligent transportation system for internet of vehicles based vehicular networks for smart cities," *Computers and Electrical Engineering*, vol. 105, p. 108543 (2023).

[2] S. Izza, M. Benssalah, and K. Drouiche, "An enhanced scalable and secure RFID authentication protocol for WBAN within an IoT environment," *Journal of Information Security and Applications*, vol. 58, p. 102705 (May 2021), doi: 10.1016/j.jisa.2020.102705.

[3] F. Hategekimana, T. J. Whitaker, M. J. Hossain Pantho, and C. Bobda, "IoT Device security through dynamic hardware isolation with cloud-Based update," *Journal of Systems Architecture*, vol. 109, p. 101827 (Oct. 2020) doi: 10.1016/j.sysarc.2020.101827.

[4] O. Abu Waraga, M. Bettayeb, Q. Nasir, and M. Abu Talib, "Design and implementation of automated IoT security testbed," *Computers & Security*, vol. 88, p. 101648 (Jan. 2020), doi: 10.1016/j.cose.2019.101648.

[5] P. Rani, M. Kumar, A. K. Das, N. Kumar, and M. Alazab, "Federated Learning-Based Misbehaviour Detection for the 5G-Enabled Internet of Vehicles," *IEEE Transactions on Consumer Electronics* (2023).

[6] M. S. Mehmood, M. R. Shahid, A. Jamil, R. Ashraf, T. Mahmood, and A. Mehmood, "A Comprehensive Literature Review of Data Encryption Techniques in Cloud Computing and IoT Environment," in 2019 8th International Conference on Information and Communication Technologies (ICICT), Karachi, Pakistan: IEEE, pp. 54–59 (Nov. 2019). doi: 10.1109/ICICT47744.2019.9001945.

[7] C. Beaman, A. Barkworth, T. D. Akande, S. Hakak, and M. K. Khan, "Ransomware: Recent advances, analysis, challenges and future research directions," *Computers & security,* vol. 111, p. 102490 (2021).

[8] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3523–3542 (2021).

[9] M. Kumar, A. Singh, A. K. Das, N. Kumar, M. Alazab, and A. Jolfaei, "Healthcare Internet of Things (H-IoT): Current Trends, Future Prospects, Applications, Challenges, and Security Issues," Electronics, vol. 12, no. 9, p. 2050 (Apr. 2023), doi: 10.3390/electronics12092050.

[10] U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat, "A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review," *Sensors,* vol. 23, no. 8, p. 4117 (Apr. 2023), doi: 10.3390/s23084117.

[11] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities," *IEEE Commun. Surv. Tutorials,* vol. 22, no. 4, pp. 2489–2520 (2020), doi: 10.1109/COMST.2020.3011208.

[12] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," *IEEE Commun. Surv. Tutorials,* vol. 22, no. 3, pp. 1686–1721 (2020), doi: 10.1109/COMST.2020.2986444.

[13] A. O. Akmandor, H. Yin, and N. K. Jha, "Smart, Secure, Yet Energy-Efficient, Internet-of-Things Sensors," *IEEE Trans. Multi-Scale Comp. Syst.,* vol. 4, no. 4, pp. 914–930 (Oct. 2018), doi: 10.1109/TMSCS.2018.2864297.

[14] K. Mandal, M. Rajkumar, P. Ezhumalai, D. Jayakumar, and R. Yuvarani, "WITHDRAWN: Improved security using machine learning for IoT intrusion detection system," *Materials Today: Proceedings*, p. S2214785320377889 (Dec. 2020), doi: 10.1016/j.matpr.2020.10.187.

[15] T. Gu, A. Abhishek, H. Fu, H. Zhang, D. Basu, and P. Mohapatra, "Towards Learning-automation IoT Attack Detection through Reinforcement Learning," in 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Cork, Ireland: IEEE, pp. 88–97 (Aug. 2020). doi: 10.1109/WoWMoM49955.2020.00029.

[16] M. M. N. Aboelwafa, K. G. Seddik, M. H. Eldefrawy, Y. Gadallah, and M. Gidlund, "A Machine-Learning-Based Technique for False Data Injection Attacks Detection in Industrial IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8462–8471 (Sep. 2020), doi: 10.1109/JIOT.2020.2991693.

[17] S. Jha, D. Prashar, H. V. Long, and D. Taniar, "Recurrent neural network for detecting malware," *Computers & Security*, vol. 99, p. 102037 (2020).

[18] A. Pinhero, M. A. Zaveri, S. Sharma, "Malware detection employed by visualization and deep neural network," *Computers & Security*, vol. 105, p. 102247 (2021).

*Received February, 2025*