# *Explainable deep learning-enabled malware attack detection for IoT-enabled intelligent transportation systems*

FULL TEXT

Journal: IEEE Transactions on Intelligent Transportation Systems
DoI: 10.1109/TITS.2025.3525505
Submission date: 27th August 2024
Acceptance date: 29th December 2024
Publication date: 16th January 2025

**Title:**

Explainable Deep Learning-Enabled Malware Attack Detection for IoT-Enabled Intelligent Transportation Systems

**Authors:**

Mohammad Wazid, *Senior Member, IEEE*
Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248 002, India (e-mail: wazidkec2005@gmail.com).

Jaskaran Singh, *Student Member, IEEE*
School of Computer Science, University of Nottingham, Nottingham, NG7 2QL, UK (e-mail: jaskaran.jsk2001@gmail.com).

Charvi Pandey, *Student Member, IEEE*
Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248 002, India (e-mail: charvipandey3@gmail.com).

R. Simon Sherratt, *Fellow, IEEE*
Department of Biomedical Engineering, University of Reading, RG6 6AY, UK (e-mail: r.s.sherratt@reading.ac.uk).

Ashok Kumar Das, *Senior Member, IEEE*
Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India, and also with the Department of Computer Science and Engineering, College of Informatics, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 02841, South Korea (e-mail: iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in).

Debasis Giri, *Member, IEEE*
Department of Information Technology, Maulana Abul Kalam Azad University of Technology, Nadia, West Bengal 741 249, India (email: debasis_giri@hotmail.com).

Youngho Park, *Member, IEEE*
School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, South Korea (e-mail: parkyh@knu.ac.kr).

Corresponding authors: Ashok Kumar Das and Youngho Park

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A3058605.

**Abstract**

The Internet of Things (IoT) has the potential to improve the complementary of communication, control, and information processing within the public transportation system. The IoT-enabled Intelligent Transportation System (ITS) ensures that automated transportation is networked and operated collaboratively. The IoT-enabled ITS has revolutionized the transportation industry by enabling the seamless integration of a wide range of devices and systems. It makes the strategic use of networked devices, sensors, and data analytics to improve transportation network efficiency, safety, and environmental friendliness. The usage of the IoT in the ITS has grown in popularity due to its capacity to improve traffic control, reduce congestion, facilitate live monitoring, and optimize transportation operations. The IoT-enabled ITS systems and devices must be protected from cyber-attacks for various reasons, including preserving sensitive data, guaranteeing privacy, preventing unauthorized access, and protecting against the risk of interruptions or manipulations. Malware attacks affect the working and performance of the deployed smart IoT devices. We propose a secure deep learning- enabled malware attack detection for IoT-enabled ITS (in short, SDLMA-IITS). The approach of explainable artificial intelligence (XAI) has been utilized for the effective detection of malware. A deep security analysis of the proposed SDLMA-IITS is presented to prove its security against various potential attacks. The comparative performance analysis of SDLMA-IITS is given with the other similar existing schemes. Finally, a practical implementation of SDLMA-IITS is provided to measure its impact on the security of the IoT-enabled ITS systems and devices.

**Index Terms**

Intelligent Transportation System (ITS), Internet of Things (IoT), Explainable artificial intelligence (XAI), malware attacks, cybersecurity, deep learning.

## I. INTRODUCTION

Cities, especially, metropolitan cities encounter a variety of difficulties as a result of their rising metropolitan population. These difficulties include, but are not limited to, high traffic congestion, decreasing air quality, an increase in road accidents, and a rapid increase in the number of private vehicles [1]. At the same time, the proportion of people who take public transit is falling. The primary cause of the problem is a lack of access to reliable public transportation infrastructure. As information technology advances, the Internet of Things (IoT) has emerged as a real-world phenomenon. The Internet of Things (IoT) has the potential to improve the complementary of communication, control, and information processing within the public transportation system. The phrase “intelligent transport systems (ITS)” refers to an upgraded version of the Vehicle Ad-hoc Network (VANET) that offers comprehensive assistance with all areas of road management operations. The IoT-enabled ITS makes the strategic use of networked devices, sensors, and data analytics to improve transportation network efficiency, safety, and environmental friendliness [2]. Its usage has become popular due to its capacity to improve traffic control, reduce congestion, facilitate live monitoring, and optimize transportation operations [3], [4]. Regardless, the development of IoT-enabled ITS is impeded by risk considerations associated with risk factors such as data confidentiality, data integrity and privacy [5]. The IoT-enabled ITS systems and devices must be protected from cyber-attacks for various reasons, including preserving sensitive data, guaranteeing privacy, preventing unauthorized access, and protecting against the risk of interruptions or manipulations. Cyber attacks (i.e., malware attacks) happening worldwide are increasing and becoming trickier daily, which calls for better ways to find and stop these attacks. Consumers indulging in illegal cyber practices are emerging and even changing their tactics. These attacks also affect the operations and functionalities of consumer IoT devices. Therefore, effective and efficient solutions (i.e., machine learning/ deep learning-based mechanisms) are required to detect and defend against these attacks in consumer IoT devices [5], [14], [15].

This paper aims to design a secure deep learning-enabled malware attack detection mechanism for IoT-enabled ITS. It looks closely at how well the proposed solution detects threats efficiently. As

these attacks increase, the data available to research also increases. The end goal is to minimize the loss and maximize the model's accuracy. The model designed for this task can be based on numerous algorithms, i.e., artificial neural network (ANN), logistic regression, and decision trees. This study has been finalized after comparing some such algorithms and then concluding with the ones that perform the best. The motive is to add to what cyber security already has and research the previously existing technologies in-depth, analyzing how they can change cyberspace.

Some of the security issues of the IoT-enabled ITS are given below [16], [17].

• Insufficient visibility: Many instances involve a lack of awareness among information technology departments regarding the utilization of the smart IoT devices by users. This poses a challenge in compiling a comprehensive inventory of all the elements that necessitate security and management [18].
• Inadequate incorporation of security protocols: Integrating smart IoT devices with security systems might be challenging or even unfeasible due to the wide variety and magnitude of these devices. There exist some deficiencies with open-source software [19].
• Open-source software: Many smart IoT devices are susceptible to security flaws and vulnerabilities. The firmware that they use is prone to various software bugs and other associated security vulnerabilities.
• Significant quantities of information: The management of data protection, administration, and monitoring poses significant challenges due to the substantial volume of data generated by smart IoT devices in the ITS environment. Therefore, essential security mechanisms, like strong user authentication and robust access control, are required [20].
• Limited security testing of smart devices in the ITS:The low emphasis on security among the majority of the IoT-enabled ITS developers results in a failure to conduct thorough vulnerability testing, which is crucial for detecting the issues of the smart IoT devices and associated systems. Hence, more security testing of smart IoT devices is required, where various fuzzing testing techniques can play an important role [21].
• Unpatched vulnerabilities: Due to the lack of security testing of smart IoT devices, a lot of devices are left with unpatched vulnerabilities, which is not good from a security point of view.
• Vulnerabilities in application programming interfaces (APIs): Exploiting insecure application programming interfaces (APIs) as ports of access to command-and-control centers is a very famous practice of hackers. The command-and-control centers serve as the primary source of various forms of attacks, including cross-site scripting attacks, SQL injection, man-in-the-middle attacks (MITM), distributed denial of service attacks (DDoS), and other forms of network breaches [22].
• Insecure passwords: Most devices in the IoT-enabled ITS environment come with default passwords that users often neglect to change. Therefore, hackers can swiftly get access to these devices. Furthermore, users may create passwords that are susceptible to being guessed. As they do not follow the required security measures [23].

### *A. Motivation*

The devices and systems in the IoT-enabled ITS environment are equipped with sensors, actuators, and software, have the capability to intelligently collect, analyze, and utilize data. This intelligence aims to enhance the efficiency of decision- making processes and automate various procedures [3], [6], [7]. Protecting IoT-enabled ITS environment from cyberattacks is crucial for several reasons: safeguarding sensitive information, ensuring privacy, preventing unauthorized access, and protecting against the risks of disruptions or manipulations in connected devices. The necessity of securing IoT-enabled ITS environment against cyber threats cannot be overstated, given their potential to cause significant harm to individuals, businesses, and critical infrastructure. The functionality and performance of the smart IoT devices is highly impacted by the malware attacks. Thus, it is essential to detect and prevent these attacks. Hence, our focus is on designing a secure and efficient deep learning-enabled mechanism for the detection of malware attacks in IoT-enabled ITS environment. The detection of malware can also become more effective through explainable artificial intelligence (XAI). Therefore, XAI has been used to make the detection of malware more effective and accurate.

*B. Research Contributions*

The research contributions of the paper are provided below.

• In this paper, we propose a secure deep learning-enabled malware attack detection for a secure IoT-enabled ITS environment (in short, SDLMA-IITS). Explainable artificial intelligence (XAI) has been used for the effective and accurate detection of malware.
• The details of the network and threat models, which belong to the proposed SDLMA-IITS are provided. These models are helpful in understanding the working and usability of the proposed SDLMA-IITS. Various important steps, like selection of dataset use of pre-processing, deployment of machine learning and deep learning algorithms, and deployment of secure authentication and key establishments, are also performed in the proposed SDLMA-IITS.
• A security analysis of the proposed SDLMA-IITS is presented to prove its security against various potential attacks.
• The comparative performance analysis of the proposed SDLMA-IITS is given with the other similar existing schemes. The proposed SDLMA-IITS outperformed the other existing schemes.
• Finally, a practical implementation of SDLMA-IITS is provided to measure its impact on real-world scenarios.

*C. Article Outline*

The remainder of the paper is organized as follows. A literature review of existing schemes is given in Section II. Section III contains different system models belonging to the proposed SDLMA-IITS. Then, the proposed SDLMA-IITS is elaborated in Section IV. Section V contains a thorough security analysis of SDLMA-IITS. Further, the practical implementation of SDLMA-IITS is conducted in Section VI. Furthermore, the comparative performance analysis of different schemes is given in Section VII. Finally, the paper is concluded in Section VIII.

## II. LITERATURE REVIEW

Gu et al. [3] presented an incentive mechanism that could potentially compensate raters for offering frank ratings. They provided a consensus model that utilized the verified delay function (VDF) in the trusted execution environment (TEE) to ensure the blockchain consortium's maximum efficiency and security. Prathiba et al. [5] introduced a cutting-edge approach known as stream-based blockchain-powered malicious node detection (BMND) for the assessment and discovery any malicious activities that might occur on the Internet of Autonomous Cars (IoAV) network, specifically autonomous vehicles (AVs) that function as nodes. Javeed et al. [6] presented an intelligent intrusion detection system (IDS) for smart consumer electronics (CE) with the help of deep learning (DL). The software-defined networking (SDN) was also deployed. They separated the data plane and the control plane. It used SDN architecture to enable reconfiguration over static network infrastructure. Then Anbalagan et al. [7] used machine learning for IDS. The stochastic gradient descent was used to improve trust evaluation in a 5G-V2X Internet of Vehicles (IoV) environment. Haghighi et al. [8] proposed another IDS. It seemed easy for automotive manufacturers for the integration of non-disruptive architecture.

Im et al. [9] presented a one-dimensional IDS that utilized WaveGAN for training by converting normal data into waveforms without any malicious activities. Their technique utilized unsupervised learning to detect attacks conducted by untrained individuals. Dib et al. [25] introduced a deep learning framework for IoT malware classification and family attribution, analyzing malware binaries to overcome the limitations of traditional machine learning classifiers that rely heavily on static and dynamic analysis. However, a key drawback of this framework was its reliance on a single representation of malware data, potentially limiting the learning process. Future research could

develop a more holistic approach, integrating various data representations and types to improve adaptability and accuracy, especially in detecting and classifying new and evolving malware families.

Qureshi et al. [26] proposed a deep learning approach for network malware analysis using LSTM, CNN, and DNN models, integrating these algorithms at the core switch of network architecture for efficient malware family classification. Such a hybrid system's complexity and high resource demands pose significant challenges. Future work could look into streamlining the architecture by developing more efficient algorithms or leveraging cloud computing resources to make the system more viable for real-world network environments.

Usman et al. [27] presented a hybrid approach using honeypots and machine learning to dynamically detect malware focusing on capturing malicious activities and analyzing malware samples in a sandbox environment. While innovative, the method's reliance on sandbox detection might lead to evasion by sophisticated malware. Future research should enhance the detection mechanisms, perhaps by incorporating advanced behavioral analysis and anomaly detection techniques, to counter evasion strategies used by advanced malware.

Ashiku and Dagli [28] introduced a CNN-based deep learning model for network intrusion detection, emphasizing convolutional operations over traditional neural network architectures. The model's complexity in tuning hyperparameters and handling tensor dimensions was a key challenge. Future work could explore automated hyperparameter optimization techniques and advanced network architectures that can more efficiently process and analyze network traffic, potentially improving the system's accuracy and adaptability to various cyber threats.

Thamilaras et al. [29] focused on developing an intrusion detection system for medical IoT devices, addressing the unique challenges of securing sensitive medical data. One obvious downside was the difficulty in striking a balance between the two competing goals of implementing robust security measures and protecting the privacy and integrity of patients' personal information. To make sure that medical IoT networks are secure and private, particularly in low-resource settings, researchers may look into developing better security schemes and machine learning algorithms.

Dutt et al. [30] presented an Intrusion Detection System (IDS) concept that was inspired by the human immune system. The model centered on two layers, which were compared to the innate and adaptive immune systems. Several noteworthy downsides include the model's complexity and the possibility of producing false positive results. Further research is needed to improve the method's precision, lower the number of inaccurate positive outcomes, and tailor the model to other network scenarios. Kasongo and Sun [31] effectively detected wireless network intrusions using a deep learning model based on feature extraction. The model's generalizability and resistance to overfitting are not particularly strong. "Graph- based convolution neural network (CNN)" was the name of the technique of Nguyen et al. [37]. Their technique was used to discover the IoT botnets. In the testing and analysis, it has been identified that their mechanism had the potential for the detection of various malware attacks. However, the malware detection accuracy of their scheme was low.

In the domain of IoT, Su et al. [38] developed a technique for identifying malicious software that is accountable for distributed denial-of-service attacks (DDoS). It was necessary to carry out the transformation to get the malware images, such as a grayscale image with a single channel, which was derived from a binary viral code. In the subsequent step, a lightweight "convolutional neural network (CNN)" was utilized to categorize the families of malicious software. They reached an accuracy of approximately 94.0% percent using the method they gave regarding classifying the malicious files. Abbas et al. [39] built multiple models for intrusion detection. It included "deep neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN)." They used the CICDIoT2023 dataset for analyses of IoT device network traffic belonging to intrusion attacks. Again Jony et al. [40] proposed a long short-term memory (LSTM) model for developing an IDS.

In delivering explainability to intrusion detection, Chinaechetam et al. [41] proposed a mechanism to counter intrusion attacks in a Metaverse environment. They used “Shapley Additive Explanations (SHAP) and local interpretable model-agnostic explanation (LIME)” explainability techniques to gain a comprehensive understanding of differentiating the behavior of an intrusion attack from legitimate benign network behavior.
Leveraging a Federated Learning framework, Abbas et al. [42] developed an edge learning Intrusion Detection System prototype with a two-client, one-server architecture. This aimed to achieve distributed training, helping to reduce the server’s computational load and incorporating privacy mechanisms.

Another approach to explainability was undertaken by Le et al. [43], who introduced an ensemble blending model to detect intrusion attacks in dynamic IoT environments using the CICIoT2023 and IoTID20 datasets. Within their framework, they employed the Local Interpretable Model-Agnostic Explanations (LIME) technique to provide human-interpretable explanations for the model’s classification of different attacks.

Jayalaxmi et al. [44] provided a survey on a hybrid frame-work proposal for an effective security model, which was applicable for intrusion detection and/or prevention. It was available in conjunction with research on risk factor analysis through a mapping approach. Their study aimed to examine the significance of various approaches, instruments, and tactics based on Artificial Intelligence (AI) that were employed in the detection and/or prevention schemes of intrusion detection in the IoT. Benkhelifa et al. [45] addressed the security problems of IoT by concentrating on the creation of technologies that could identify intrusions in the IoT. Their study offered a thorough analysis of the latest intrusion detection systems (IDSs) designed for the IoT. Their specific emphasis was on various architectural approaches. Subsequently, a proposal for prospective advancements in the IoT intrusion detection systems was formulated and assessed. Their demonstration illustrated how the intrinsic attributes of conventional methods lead to inadequate coverage of the IoT field, making them unsuitable for their uses in IoT security. Arisdakessian et al. [46] offered a thorough examination of the IDS methods for the IoT ecosystem. This ecological framework encompassed smart IoT devices, along with the intercommunications that transpired among the tiers of cloud computing, fog computing, and IoT. Sisodia et al. [47] presented “TWINKLE, ” a security architecture that operated in two modes, i.e., regular mode and attentive mode. In normal mode, the IoT network consumed fewer resources, while in attentive mode, it only consumed additional resources when suspicious activity was discovered. Irfan Simsek [48] presented an innovative methodology that facilitates authentication, authorization, access control, and key exchange for secure device-to-device communication. The objective of their approach was to streamline the administration of cloud/fog-based and blockchain-based technological systems. Further, their scheme integrated zero-knowledge and identity-based strategies to address the demands of IoT security while also fulfilling these security objectives in a harmonious manner.

The summary and analysis of existing techniques are provided in Table I.

## III. SYSTEM MODEL

In this section, the network and threat models are associated with the proposed SDLMA-IITS. The details are given below.

### *A. Network Model*

The network model of the proposed SDLMA-IITS is given in Fig. 1. This diagram shows the flow of data from acquisition to analysis and decision-making at the cloud servers, which are used for malware sample analysis, detection, and predictions. The network model consists of the arrangement of smart IoT devices, users, and cloud servers connected to the network [10], [11]. The cloud servers constantly communicate with the devices and then analyze them. The IoT-enabled ITS devices and systems are prone to various cyber-attacks [5]. Implementing a robust intrusion detection system is

necessary to secure the network from potential threats and invasion. The system proposed in this study comprises multiple steps, starting with data pre-processing. This includes collecting data from the above-mentioned devices. The data pre-processing is also responsible for data quality issues by dealing with missing values through data cleaning. The next step is for feature selection to identify relevant attributes for detecting malware [12]. After that, data transformation is done by normalizing numerical values or encoding certain variables. Then, certain features are added or dropped according to their relevance. After this, the deployed machine learning/deep learning algorithms are used for malware detection [13]. These algorithms learn from the pre-processed data to accurately predict malware entities. The deployed algorithms aim to detect and neutralize threats posed by attackers, strengthening the consumer IoT system's cybersecurity defenses [14].

### *B. Threat Model*

In this paper, we follow the widely-accepted DolevYao (DY) model. Under this model, the parties communicating communicate with one another through the general open channel, which is understood to be the Internet. It is not safe to use this channel. Consequently, the possible attacker is afforded a few opportunities to delete, view, or edit the messages that have been trading hands [33]. In addition to that, we have adhered to the principles that are outlined in the CK-adversary model [34]. The attacker is said to possess all of the capabilities that are present in the DY model. Furthermore, it is worth noting that the attacker has the capability to steal the session states and the information that is linked with them [49], [50]. In other words, session keys can be disclosed to the attacker if they are not handled appropriately. Furthermore, IoT-enabled ITS devices and systems are also vulnerable to various malware (i.e., spyware, ransomware, trojan horse, rootkit, etc.). Therefore, threats caused by malware attacks should also be considered to design an effective security mechanism [15], [24].

TABLE I
SUMMARY OF EXISTING TECHNIQUES: METHODS APPLIED, THEIR DRAWBACKS/LIMITATIONS AND FUTURE WORKS

| **Technique** | **Methods applied** | **Drawbacks/Limitations** | **Possible future works** |
|---|---|---|---|
| Dib *et al.* [25] | Deep learning framework for IoT malware classification, analyzing malware binaries. | Reliance on a single representation of malware data, potentially limiting learning. | Develop a more holistic approach integrating various data representations. |
| Qureshi *et al.* [26] | Deep learning approach using LSTM, CNN, DNN models for network malware analysis. | Complexity and high resource demands of the hybrid system. | Streamline the architecture and develop more efficient algorithms or leverage cloud computing. |
| Usman *et al.* [27] | Hybrid approach using honeypots and machine learning for dynamic malware detection. | Reliance on sandbox detection may lead to evasion by sophisticated malware. | Enhance detection mechanisms with advanced behavioral analysis and anomaly detection. |
| Ashiku and Dagli [28] | CNN-based deep learning model for network intrusion detection. | Complexity in tuning hyperparameters and handling tensor dimensions. | Explore automated hyperparameter optimization techniques and advanced net- work architectures. |
| Thamilaras *et al.* [29] | Intrusion detection system for medical IoT devices, securing sensitive medical data. | Balancing between ensuring robust security and maintaining data privacy. | Develop advanced encryption techniques and machine learning algorithms for medical IoT networks. |
| Dutt *et al.* [30] | IDS model inspired by the human immune system, focusing on two-layer defense. | Complexity of the model and potential for false positives. | Refine the algorithm to enhance accuracy, reduce false positives, and adapt to various network scenarios. |
| Kasongo and Sun [31] | Deep learning model with feature extraction for detecting intrusions in wireless networks. | Model's ability to generalize and avoid overfitting. | Explore advanced feature selection and extraction techniques to improve model adaptability. |
| Nguyen *et al.* [37] | Graph-based convolution neural network (CNN) | Accuracy value is low. | Accuracy should be improved. Proper security analysis should be provided. |
| Abbas *et al.* [39] | Multiple deep learning models (DNN, CNN, and RNN) | Performance metrics has a lot of potential for improvement | Feature Extraction could be improved. Proper security analysis should be provided. |
| Jony *et al.* [40] | Long Short-term memory models | The proposed scheme is limited and may not work effectively with biased and complex data | Explore more models and feature extraction to improve scheme's effectiveness |
| Chinaechetam *et al.* [41] | Dense Neural Network | No discussion of Metaverse cyber attack threat model | Proper security analysis should be provided. The proposed scheme's connection to the metaverse should be clarified and worked upon more. |
| Abbas *et al.* [42] | Federated edge learning with Dense Neural Networks | Federated learning's benefit in Computational load was not evaluated | Privacy proper security analysis should be provided establishment has to be practically tested with validation |
| Le *et al.* [43] | Ensemble blending model | No proper justification for machine learning model selection. Ensemble still contains machine learning models as solo components which are less efficient at feature extraction than deep learning models | Ensemble can be built with deep learning models to make more domain adaptive and generalized framework |

## IV. SDLMA-IITS: THE PROPOSED SCHEME

The proposed SDLMA-IITS is designed through various steps, i.e., selection of dataset use of pre-processing, deployment of machine learning and deep learning algorithms, and deployment of secure authentication and key establishments.

### *A. Dataset Selection*

To practically validate our scheme, we selected CICIoT2023 [32] as our dataset on which we evaluated the performance of our scheme. This IoT attack dataset captures 33 different attacks, comprising 7 major categories and benign data. All data were collected from 105 IoT devices. For our evaluation purposes, we consider the problem to be a binary (benign/attack) scenario. The dataset provides important insights into the behavior of cyber threats, including the Duration of the packet's flow, Fin flag value, Rate of outbound packets, Protocol type, and Header length. These features offer information for analyzing and understanding cyber-attack methods on practical consumer-centric IoT devices. This dataset is a key resource for training and testing machine learning and deep learning algorithms for detecting and preventing cyber threats. The study performs various algorithms on the dataset, including ANN, CNN, CNN, and LSTM hybrid model, and CNN-Gated recurrent unit (GRU) ensemble model.

### *B. Pre-processing and Dataset Exploration*

There is a huge amount of data around malware to analyze. This data can be in the form of tables, charts, or graphs. One such tabular dataset has been considered for studying the malicious entities' patterns. Data preprocessing is the most important step before applying any algorithms to it. The biggest reason for this is the data quality, which includes missing values, lack of consistency, or wrong entries. Solving these issues makes the training and testing foolproof and thus leads to better results. Data preprocessing also plays an important role in selecting only the necessary features. This makes it easy for the model to get trained and decreases the chances of data overfitting. The whole process involves cleaning, making relevant changes, adding or dropping some values, and then splitting to make it suitable for training and testing. The steps are also elaborated in Algorithm 1.

We also undertook data exploration to obtain a better visualization of the processed dataset. Fig. 2 provides an elaborate visualization of the 34 attack and benign classes present in the dataset. After applying categorical encoding, we calculated the correlation between all the input features and visualized it as a Heatmap in Fig. 3. From the inference, we observe that ("min," "avg," "tot size," "number," "radius," "syn_flag_number," "psh_flag_number," and "ece_flag_number") have the most positive correlation among themselves.

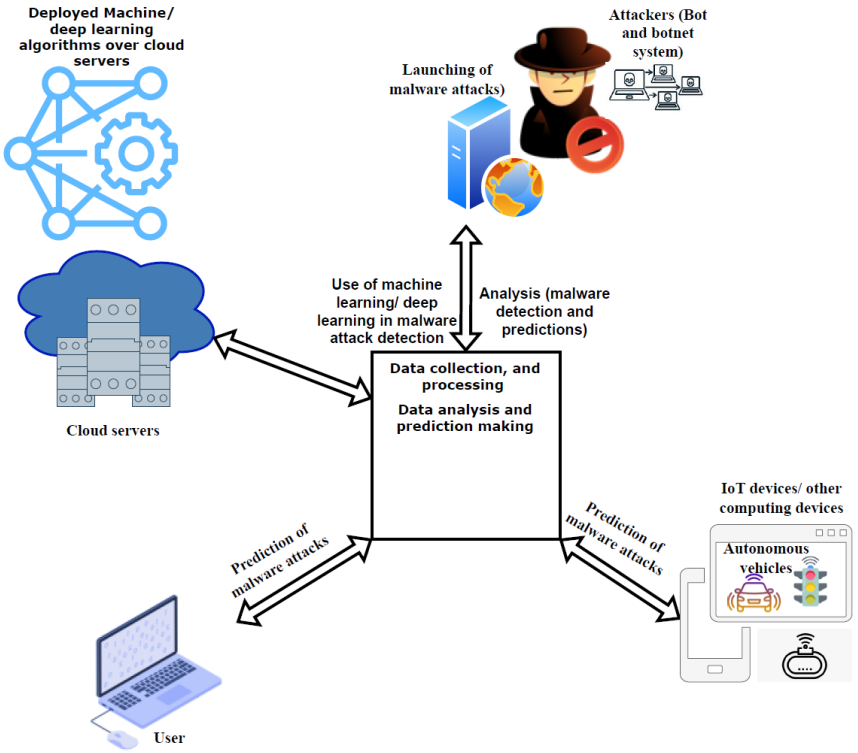


Fig. 1. Network model of the proposed SDLMA-IITS

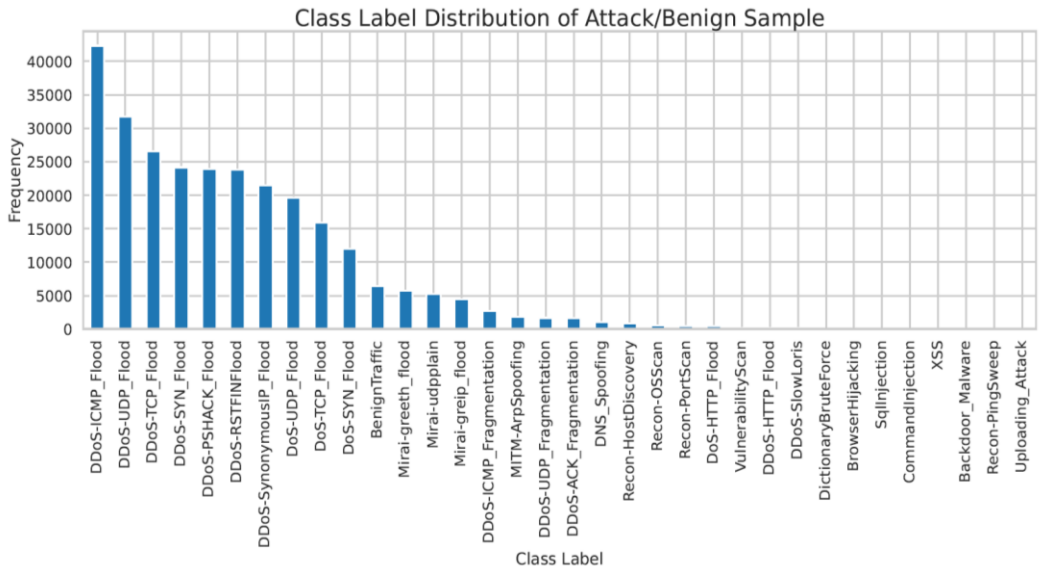


Fig. 2. Class Label Distribution of Attack/Benign Samples

Further exploration and analysis led to further interesting observations about the dataset. Fig. 4 showcases the protocol usage frequency in the traffic. Internet Protocol (IP) and Logical Link Control (LLC) protocol's frequency is the highest, followed by Transmission Control Protocol (TCP) as having third most frequency.

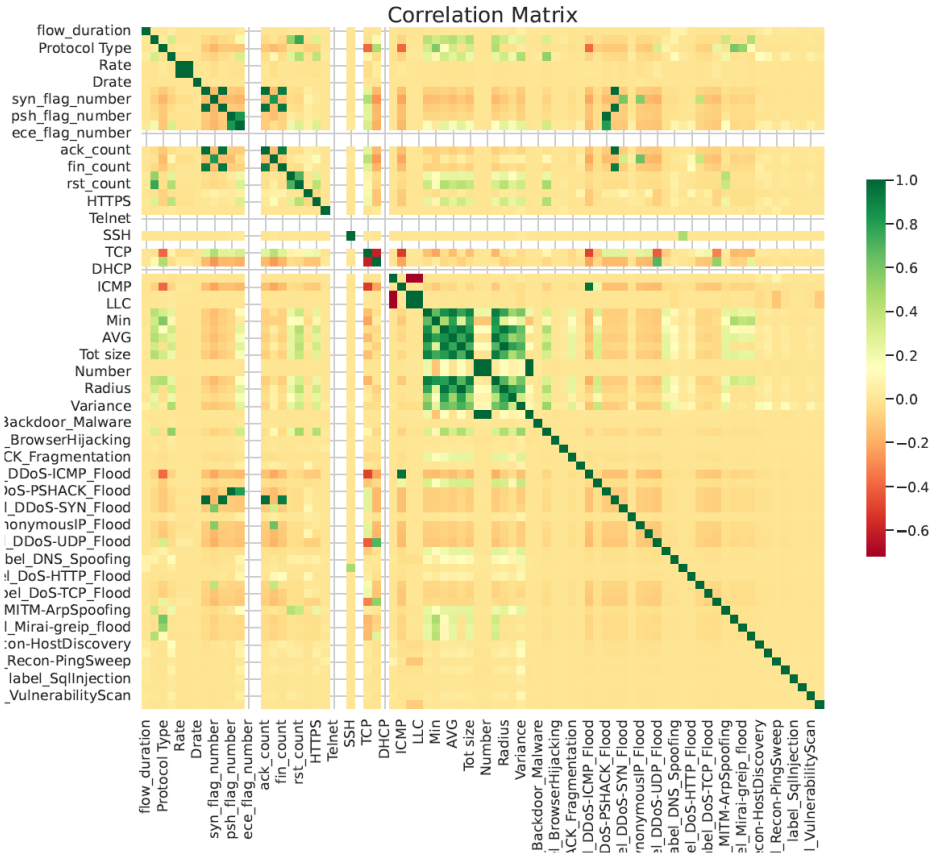


Fig. 3. Correlation matrix heatmap of input features

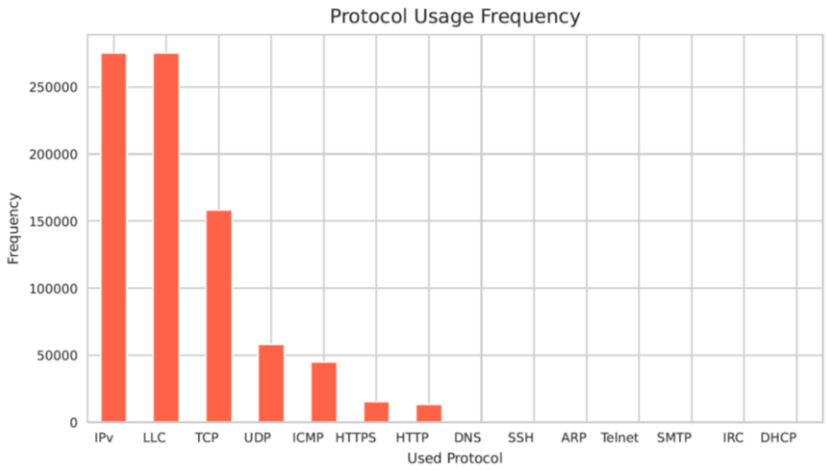


Fig. 4. Protocol Usage Frequency of Network Samples

*C. Machine Learning and Deep Learning Algorithms Deployment*

The following machine learning and deep learning algorithms have been utilized for malware detection. All these machine learning algorithms need to be deployed at the resource-rich cloud servers as they have enormous amounts of computation, storage, and communication capabilities, which are essentially required for the task of malware detection. The outcome of this phase is in the form of the prediction about the detected malicious malware programs if they exist in the system. The details of the machine learning and deep learning algorithms according to their uses are provided below.

**Algorithm 1** Preprocessing

**for** each attribute in the dataset **do**
  Eliminate the outliers and deal with the values that are missing.
  **if** The values of the attributes are not numeric **then**
    Convert them into numerical flags.
  **else**
    Continue with the existing values.
  **end if**
  **if** The values of the property are redundant or the attribute only has a single consistent value **then**
    The attribute should be removed.
  **else**
    Continue with the existing values.
  **end if**
  **if** It is required **then**
    Normalize the data by inserting appropriate values and formats.
  **else**
    Continue with the existing values and formats.
  **end if**
**end for**

*1) Logistic Regression:* When it comes to classification, logistic regression is a straightforward and efficient approach. It creates a model of the relationship between the variables that are dependent and unrelated. This algorithm is simple to comprehend and put into practice. This leads to its widespread adoption and makes it a good choice for complex binary classification problems. The required steps of logistic regression are given in Algorithm 2.

**Algorithm 2** Deployed logistic regression algorithm

**for** each attribute in the dataset **do**
  Import necessary libraries, like Scikit-learn.
  Import the dataset.
  Perform data preprocessing, a. handle missing values, b. encode categorical data, and c. feature scaling.
  **for** $k = 1$ to $10$ **do**
    k-fold cross-validation.
    Split dataset into $90\%$ training and $10\%$ test set.
  **end for**
  Initialize the logistic regression model and set its parameters: a. solver, b. regularization strength.
  Fit the model with the training set.
  Make predictions on the test set.
  **for** Evaluation and assessment **do**
    Compute accuracy and F1-score comparing predicted output and ground truth.
  **end for**
**end for**

*2) Artificial Neural Networks (ANN):* An artificial neural network (ANN) is a strong machine-learning model that is made up of layers of neurons that are connected and process information. These are capable of managing intricate data patterns. The steps of the deployed artificial neural networks (ANN) algorithm are elaborated in Algorithm 3.

**Algorithm 3** Deployed artificial neural networks (ANN) algorithm

**for** each attribute in the dataset **do**
  Import necessary libraries: TensorFlow, and Keras.
  Import the dataset.
  Perform data preprocessing, a. handle missing values, b. encode categorical data, and c. feature scaling.
  **for** $k = 1$ to $10$ **do**
    k-fold cross-validation.
    Split dataset into $90\%$ training and $10\%$ test set.
  **end for**
  Define ANN model, a. input layer, b. hidden layers, c. dropout layer, d. output layer, and e. activation functions.
  Compile the ANN model, a. select optimizer, b. loss function, and c. metrics.
  Fit the model on the training set.
  Make predictions on the test set.
  **for** Evaluation and assessment **do**
    Compute accuracy and F1-score comparing predicted output and ground truth.
  **end for**
**end for**

*3) Convolution Neural Networks (CNN):* Convolution neural network (CNN) is a very strong method of deep learning. It uses the convolution filters. That helps to extract different features. The collected temporal information is further used in conjunction with neurons of artificial neural network layers. The required steps of the CNN algorithm are given in Algorithm 4.

**Algorithm 2** Deployed logistic regression algorithm

**for** each attribute in the dataset **do**
  Import necessary libraries, like Scikit-learn.
  Import the dataset.
  Perform data preprocessing, a. handle missing values, b. encode categorical data, and c. feature scaling.
  **for** $k = 1$ to $10$ **do**
    k-fold cross-validation.
    Split dataset into $90\%$ training and $10\%$ test set.
  **end for**
  Initialize the logistic regression model and set its parameters: a. solver, b. regularization strength.
  Fit the model with the training set.
  Make predictions on the test set.
  **for** Evaluation and assessment **do**
    Compute accuracy and F1-score comparing predicted output and ground truth.
  **end for**
**end for**

*4) Convolution Neural Network Long-short Term Memory (CNN-LSTM):* Long Short-Term Memory (LSTM) is a specialized version of the Recurrent Neural Network. It includes dedicated memory cells for maintaining sequential information. Consequently, they are highly optimized for extracting temporal dependencies from the input data. These powerful models can be combined with CNN layers to create a hybrid CNN-LSTM model, enabling the extraction of features and the gathering of information about the data's nature in a more generalized manner. The steps of the CNN-LSTM algorithm are elaborated in Algorithm 5.

**Algorithm 5** Deployed Convolution Neural Network Long-short Term Memory (CNN-LSTM) algorithm

**for** each attribute in the dataset **do**
  Import necessary libraries: TensorFlow, and Keras.
  Import the dataset.
  Perform data preprocessing, a. handle missing values, b. encode categorical data, and c. feature scaling.
  **for** $k = 1$ to $10$ **do**
    k-fold cross-validation.
    Split dataset into $90\%$ training and $10\%$ test set.
  **end for**
  Define CNN model, a. input layer, b. convolution layer filter size, c. pooling layer, d. flatten layer, e. dropout layer, f. hidden dense layers, g. output layer, and h. activation functions.
  Define LSTM layer, a. resizing layer. LSTM layer size, c. flatten layer, d. dropout layer, and e. activation functions.
  Compile the CNN-LSTM model, a. select optimizer, b. loss function, and c. metrics.
  Fit the model on the training set.
  Make predictions on the test set.
  **for** Evaluation and assessment **do**
    Compute accuracy, and F1-score comparing predicted output and ground truth.
  **end for**
**end for**

*5) Ensemble Model (CNN-GRU || CNN-LSTM):* To leverage the benefits of multiple models, we constructed an ensemble model comprising two hybrid models: CNN-GRU and CNN-LSTM. This ensemble model extracted features and relations from the input network traffic separately through the two submodels, yielding two output vectors. These output vectors were subsequently fused together using a concatenation function to obtain a combined output. Finally, this output vector was passed through multiple dense layers to generate a final prediction. By harnessing the strengths of multiple models, we were able to capture various aspects of the data, resulting in reduced variance, improved generalization, and state-of-the-art performance. The steps of the Ensemble model (CNN-GRU || CNN-LSTM) algorithm are elaborated in Algorithm 6.

**Algorithm 6** Deployed Ensemble Model (CNN-GRU || CNN-LSTM) algorithm

**for** each attribute in the dataset **do**
  Import necessary libraries: TensorFlow, and Keras.
  Import the dataset.
  Perform data preprocessing, a. handle missing values, b. encode categorical data, and c. feature scaling.
  **for** $k = 1$ to 10 **do**
    k-fold cross-validation.
    Split dataset into 90% training and 10% test set.
  **end for**
  Define CNN-GRU model $\alpha$, a. input layer, b. convolution layer filter size, c. GRU layer size d. pooling layer, e. flatten layer, f. dropout layer, g. hidden dense layers, h. output layer, and i. activation functions.
  Define CNN-LSTM model $\beta$, a. input layer, b. convolution layer filter size, c. LSTM layer size d. pooling layer, e. flatten layer, f. dropout layer, g. hidden dense layers, h. output layer, and i. activation functions.
  Define Fusion layer and Output model, which will fuse both the output layers from model $\alpha$ and $\beta$ using concatenation $\theta$ function and connect to more dense layers to get a combined output layer.
  Compile the Ensemble Model model, a. select optimizer, b. loss function, and c. metrics.
  Fit the model on the training set.
  Make predictions on the test set.
  **for** Evaluation and assessment **do**
    Compute accuracy, and F1-score comparing predicted output and ground truth.
  **end for**
**end for**

*D. Secure Authentication and Key Establishment*

In the proposed SDLMA-IITS, authentication and key establishments are provided among the various cloud servers and computing devices. This procedure is required due to their secure data transmissions. To accomplish secure authentication and key establishments, some standard mechanisms have been presented by Wazid et al. [35], and Srinivas et al. [36] that can be used. After completing the required steps of an authentication and key establishment procedure, the cloud server $SRV_i$ and computing device $CD_j$ establish the session key say $SK_{SRVi,CDj}$ ($=SK_{CDj,SRVi}$) for their secure communication. For example, $CD_j$ can send its data $Dt_{CDj}$ to $SRV$j after performing the encryption with $SK_{CDj\ ,SRVi}$, i.e., $MSG1_{CDj\ ,SRVi} = E_{SKCDj,\ SRVi}(D_{tCDj})$. At the arrival of MSG1CDj,SRVi , SRVi performs decryption as DSKSR ,CD (MSG1CDj,SRVi ) and retrieves DtCDj for its further processing and storage.

*E. Process flow diagram of the proposed SDLMA-IITS*

The process flow diagram gives a view of the flow of execution of the various phases of a scheme. By the following these phases the detection of malware programs is performed in the proposed SDLMA-IITS. The flow of execution of multiple processes operating as part of the proposed SDLMA- IITS is given in Fig. 5. The details are also given below.

• Registration of IoT devices takes place by initializing the ad-hoc device. There is also the authentication between smart IoT devices and the cloud server, as well as with the daemon engine.
• The attacker tries to attack smart IoT device by manipulation, interruptions, unauthorized access, or phishing attempts. Simultaneously the proxy device collects malware-infected data posing as real IoT devices.

• Network traffic files are sniffed and extracted continuously from the IoT devices by the daemon engine. Meanwhile, the data collected by honeypotting by the proxy devices are given to the deployed model for its continuous finetuning.
• Extracted network traffic data is preprocessed using deployed scripts by the daemon engine.
• Daemon engine delivers the preprocessed data to the cloud server for predictive analysis.
• The pre-processed data is evaluated, and the behavior is analyzed by the deployed ML and DL models.
• The generated prediction report and findings are sent back to the Daemon engine.
• Based on the severity of the threat, the consumer is alerted, and the results are conveyed

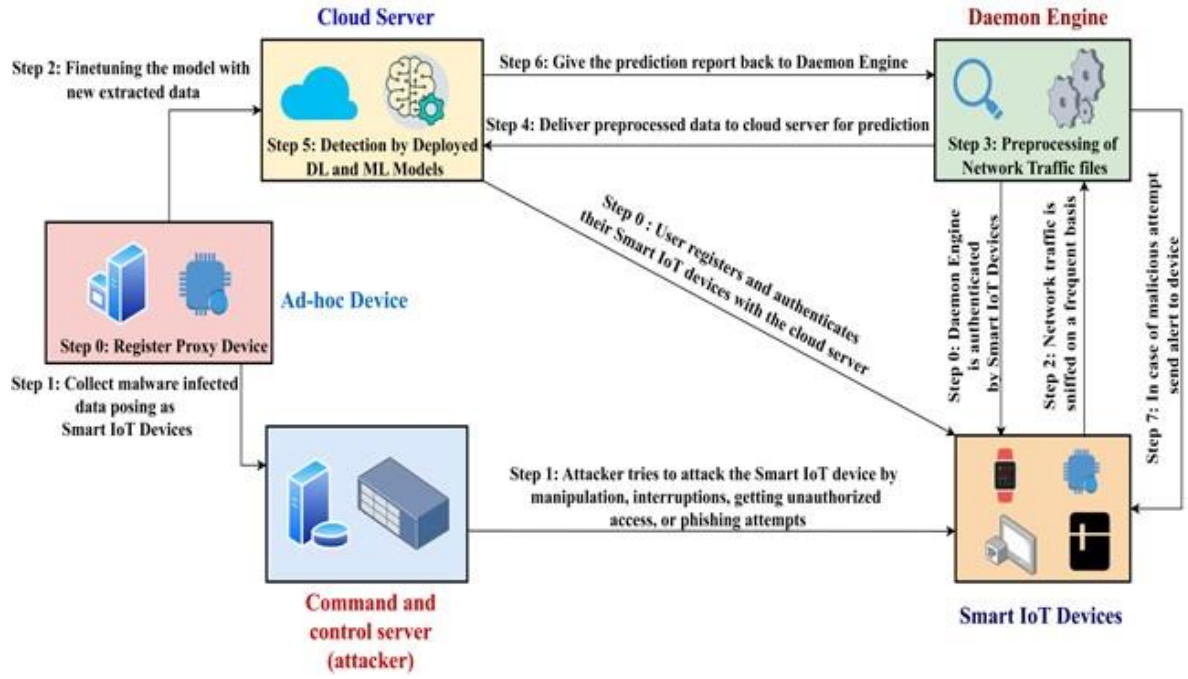


Fig. 5. Process flow diagram of the proposed SDLMA-IITS

## V. SECURITY ANALYSIS

In this section, we provide the security analysis of the proposed SDLMA-IITS.

*A. Security against Malware Attacks*

The proposed SDLMA-IITS has the ability to defend against malware attacks through the mechanism given in Section IV-C. Due to the provided steps, i.e., “use of pre-processing” and “deployment of machine learning and deep learning algorithms,” SDLMA-IITS detects and predicts the potential malware if it exists in the system

*B. Security against Other Potential Attacks*

Due to the presence of the “deployment of secure authentication and key establishments” phase given in Section IV-D, SDLMA-IITS has the ability to defend against other potential attacks, i.e., it can mitigate replay attacks due to the use of freshly generated timestamp values, which are verified at the receivers end. Moreover, in all exchanged messages, we use “short-term secret values, i.e., random nonce/ number values” and “long-term secret values, i.e., various identities and secret keys.” These

values are also used in the creation of the session keys. Because of that, we get the secured session keys without the possibility of any disclosure attacks on them. It also provides different messages in different sessions. Therefore, traceability of the messages is not possible. In all messages, there is the provision to use pseudo identities in place of original identity. Hence, the entire communication becomes anonymous. These mechanisms also protect SDLMA-IITS against other potential attacks, i.e., man-in-them-middle at- tacks, impersonation, stolen verifier, etc. As per the given discussion, it is clear that SDLMA-IITS has the ability to defend against potential attacks on the system.

## VI. PRACTICAL IMPLEMENTATION

In this section, we provide the details of the practical implementation of the proposed SDLMA-IITS.

### *A. Simulation Environment and Setting*

In the practical implementation of SDLMA-IITS, we have used the following simulation environment and setting. The programming platform was Jupyter Notebook, along with the Python programming language. The packages/libraries were TensorFlow, Keras, NumPy, Pandas, Scikit-learn, Matplotlib and Seaborn. The details of simulation parameters, along with their values, are given in Table II.

TABLE II
SIMULATION PARAMETERS AND THEIR VALUES

| **Parameter** | **Explanation** |
|---|---|
| Programming platform | Jupyter notebook |
| Programming language | Python |
| Dataset | Malware dataset [32] |
| Packages/libraries | TensorFlow, Keras, NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn |

### *B. Results and Discussion*

The results from the evaluation of different models for malware detection show different levels of effectiveness. CNN-LSTM hybrid model performed the best among the four tested algorithms with an accuracy of 99.28% and an F1 score of 0.986. This showcases the Hybrid model's ability to classify and identify malicious entity patterns accurately in a generalized manner. The plain CNN model showed a similar result with a 99.25% accuracy and an F1 score of 0.974. The baseline ANN model also delivered great performance with an accuracy of 97.63% and an F1 score of 0.95. The obtained results are given in Table III, Fig. 6, and Fig. 7.

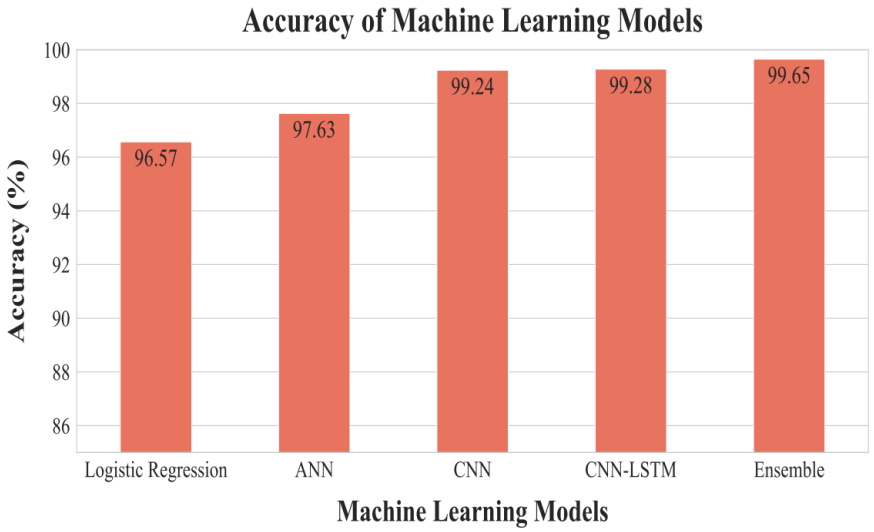


Fig. 6. Performance comparisons of various deployed algorithms: Accuracy values

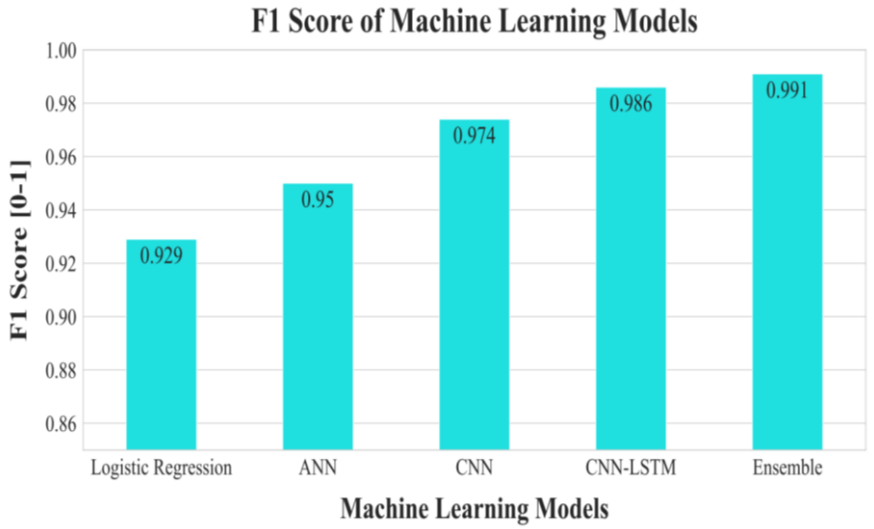


Fig. 7. Performance comparisons of various deployed algorithms: F1-score values

*C. Explainable AI*

In a bid to infer understanding from the built models and extract valuable findings about the features, we used an explainable artificial intelligence (XAI) algorithm on the deployed Machine learning models. In particular, we used the “SHapley Additive exPlanations (SHAP) algorithm” for explainability on top of a random forest-based model and generated a summary plot. The plot visualized in Fig. 8 showcases the feature importance for the top features based on the nature of the sample (attack/ benign). We can infer that the “rst_count” and “IAT” features have almost the maximum importance in the sample prediction by the model. It was followed by “urg_counter,” i.e., the number of packets with urg flag set in the same flow. Interestingly, the number of packets with the urg flag set in the same flow (Strate) has the lowest impact among the top features on the model’s prediction. In this way, XAI helps us to make the detection of malware programs more accurate.

## VII. COMPARATIVE STUDY

This section provides a comparative performance analysis of the existing and proposed SDLMA-IoT on the CICIoT2023. The accuracy values of the schemes of Abbas et al. [39], Jony et al. [40], Chinaechetam et al. [41], Abbas et al. [42], and Le et al. [43] are 96.52%, 98.75%, 99.10%, 99.15%, and 99.51% respectively. Whereas the proposed SDLMA-IITS has achieved an accuracy of 99.63%, which is better than the other existing schemes. The comparison of accuracy values of various schemes is given in Table IV.

TABLE III
OBTAINED RESULTS

| Method | Accuracy | F1-Score |
|---|---|---|
| Logistic Regression | 96.57% | 0.929 |
| ANN | 97.63% | 0.95 |
| CNN | 99.24% | 0.974 |
| CNN-LSTM | 99.28% | 0.986 |
| Ensemble (CNN-GRU ‖ CNN-LSTM) | 99.65% | 0.991 |

TABLE IV
COMPARISON OF ACCURACY VALUES AMONG VARIOUS SCHEMES

| Method | Accuracy |
|---|---|
| Abbas *et al.* [39] | 96.52% |
| Jony *et al.* [40] | 98.75% |
| Chinaechetam *et al.* [41] | 99.10% |
| Abbas *et al.* [42] | 99.15% |
| Le *et al.* [43] | 99.51% |
| Proposed SDLMA-IITS | 99.65% |

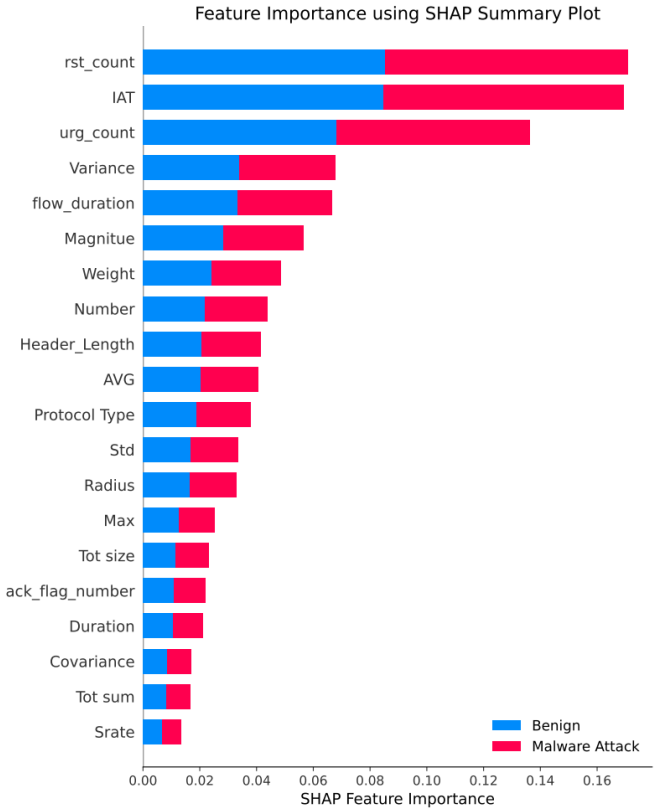


Fig. 8. Feature importance using SHAP summary plot in benign and malware samples

## VIII. CONCLUSION AND FUTURE WORK

Malware attacks can disrupt the functioning and performance of devices and systems in the IoT-enabled ITS environment. Detecting and mitigating such attacks is crucial. We presented a secure deep learning-based mechanism for detecting malware attacks in IoT-enabled ITS environment (in short, SDLMA-IITS). XAI was used to detect malware programs effectively and accurately. A security analysis was provided to validate its effectiveness against potential threats. Comparative performance analysis with existing schemes demonstrated the proposed SDLMA-IITS's superiority in essential performance parameters. Finally, we provided the practical implementation of SDLMA-IITS to assess its impact on the security of consumer IoT devices.

In the future, we would like to add more functionality features to the presented scheme. We also want to enhance the accuracy of the presented scheme further.

## REFERENCES


[1] S. Geetha and D. Cicilia, "IoT enabled intelligent bus transportation system," 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2017, pp. 7-11, Coimbatore, India.
[2] Karthikeyan H, Usha G. "A secured IoT-based intelligent transport system (IoT-ITS) framework based on cognitive science," Soft comput. 2023 May 15:1-11. doi: 10.1007/ s00500- 023- 08410-7.
[3] C. Gu, B. Ma and D. Hu, "A Dependable and Efficient Decentralized Trust Management System Based on Consortium Blockchain for Intelligent Transportation Systems," in IEEE Transactions on Intelligent Transportation Systems, 2024, doi: 10.1109/ TITS. 2024. 3443909.
[4] M. S. Peelam, Naren, M. Gera, V. Chamola and S. Zeadally, "A Review on Emergency Vehicle Management for Intelligent Transportation Systems," in IEEE Transactions on Intelligent Transportation Systems, 2024, doi: 10.1109/ TITS. 2024. 3440474.
[5] S. B. Prathiba, P. Murali, R. S. Moorthy, D. K. Anandhan, A. K. Selvaraj and J. J. P. C. Rodrigues, "A Blockchain-Powered Malicious Node 54 Detection in Internet of Autonomous Vehicles," in IEEE Transactions on Intelligent Transportation Systems, 2024, doi: 10.1109/ TITS. 2024. 3433480.
[6] D. Javeed, M. S. Saeed, I. Ahmad, P. Kumar, A. Jolfaei and M. Tahir, "An intelligent intrusion detection system for smart consumer electronics network," IEEE Transactions on Consumer Electronics, vol. 69, no. 4, pp. 906-913, 2023.
[7] S. Anbalagan, G. Raja, S. Gurumoorthy, D. Suresh R and K. Ayyakannu, "Blockchain assisted hybrid intrusion detection system in autonomous vehicles for industry 5.0," IEEE Transactions on Consumer Electronics, vol. 69, no. 4, pp. 881-889, 2023.
[8] M. Sayad Haghighi, F. Farivar, A. Jolfaei, A. B. Asl and W. Zhou, "Cyber attacks via consumer electronics: studying the threat of covert malware in smart and autonomous vehicles," IEEE Transactions on Consumer Electronics, vol. 69, no. 4, pp. 825-832, 2023.
[9] H. Im, D. Kim and S. Lee, "Listening to CAN: anomaly detection to enhance in-vehicle network security," IEEE International Conference on Consumer Electronics-Berlin (ICCE-Berlin), pp. 172-175, 2023.
[10] A. Chehri, G. Jeon, F. Rivest, and H. T. Mouftah, "Evolution and trends in artificial intelligence of things security: when good enough is not good enough!," IEEE Internet of Things Magazine, vol. 5 (3), pp. 62- 66, September 2022.
[11] M. Shen, A. Gu, J. Kang, X. Tang, X. Lin, L. Zhu, and D. Niyato, "Blockchains for artificial intelligence of things: a comprehensive sur- vey," IEEE Internet of Things Journal, vol. 10 (16), pp. 14483-14506, 2023.
[12] H. Bao, Y. Zhao, X. Zhang, G. Wang, J. Duan, R. Tian, J. Men, and M. Zhang, "A probabilistic and distributed validation framework based on blockchain for artificial intelligence of things," IEEE Internet of Things Journal, vol. 11 (1), pp. 17-28, 2024.
[13] M. Dai, Z. Su, R. Li, Y. Wang, J. Ni and D. Fang, "An edge-driven security framework for intelligent internet of things," IEEE Network, vol. 34 (5), pp. 39-45, 2020.

[14] M. Adil, M. A. Jan, Y. Liu, H. Abulkasim, A. Farouk and H. Song, “A systematic survey: security threats to UAV-aided IoT applications, taxonomy, current challenges and requirements with future research directions,” IEEE Transactions on Intelligent Transportation Systems, vol. 24 (2), pp. 1437-1455, Feb. 2023.
[15] M. Bouzidi, N. Gupta, F. A. Cheikh, A. Shalaginov and M. Derawi, “A novel architectural framework on IoT ecosystem, security aspects and mechanisms: a comprehensive survey,” IEEE Access, vol. 10, pp. 101362-101384, 2022.
[16] M. Adam, M. Hammoudeh, R. Alrawashdeh and B. Alsulaimy. “A Survey on Security, Privacy, Trust, and Architectural Challenges in IoT Systems,” IEEE Access, doi: 10.1109/ACCESS.2024.3382709.
[17] N. T. Y. Huan and Z. A. Zukarnain. “A Survey on Addressing IoT Security Issues by Embedding Blockchain Technology Solutions: Re- view, Attacks, Current Trends, and Applications,” IEEE Access, doi: 10.1109/ACCESS.2024.3378592.
[18] P. Sun, S. Shen, Y. Wan, Z. Wu, Z. Fang and X. -z. Gao. “A Survey of IoT Privacy Security: Architecture, Technology, Challenges, and Trends,” IEEE Internet of Things Journal, doi: 10.1109/JIOT.2024.3372518.
[19] N. F. Karagiorgos, S. G. Stavrinides, C. de Benito, S. Nikolaidis and R. Picos, “Unconventional Security for the IoT: Hardware and Software Im- plementation of a Digital Chaotic Encrypted Communication Scheme,” IEEE Internet of Things Journal, doi: 10.1109/JIOT.2024.3371091.
[20] A. Alrawais, A. Alhothaily, C. Hu and X. Cheng. “Fog Computing for the Internet of Things: Security and Privacy Issues,” IEEE Internet Computing, vol. 21, no. 2, pp. 34-42, 2017.
[21] J. Zhou, Z. Cao, X. Dong and A. V. Vasilakos. “Security and Privacy for Cloud-Based IoT: Challenges,” IEEE Communications Magazine, vol. 55, no. 1, pp. 26-33, 2017.
[22] Y. Yu, Y. Li, J. Tian and J. Liu. “Blockchain-Based Solutions to Security and Privacy Issues in the Internet of Things,” IEEE Wireless Communications, vol. 25, no. 6, pp. 12-18, 2018.
[23] B. Nour, K. Sharif, F. Li and Y. Wang. “Security and Privacy Challenges in Information-Centric Wireless Internet of Things Networks,” in IEEE Security & Privacy, vol. 18, no. 2, pp. 35-45, 2020.
[24] J. Srinivas, A. K. Das, M. Wazid and A. V. Vasilakos, “Designing secure user authentication protocol for big data collection in IoT-based intelligent transportation system,” IEEE Internet of Things Journal, vol. 8 (9), pp. 7727-7744, 2021.
[25] M. Dib, S. Torabi, E. Bou-Harb and C. Assi, “A multi-dimensional deep learning framework for IoT malware classification and family attribution,” IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1165-1177, 2021.
[26] S. Qureshi, J. He, S. Tunio, N. Zhu, F. Akhtar, F. Ullah, A. Nazir, and A. Wajahat, “A hybrid DL-based detection mechanism for cyber threats in secure networks,” IEEE Access, vol. 9, pp. 73938-73947, 2021.
[27] N. Usman, S. Usman, F. Khan, M. A. Jan, A. Sajid, M. Alazab, and P. Watters, “Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics,” Future Generation Computer Systems, vol. 118, pp. 124-141, 2021.
[28] L. Ashiku, and C. Dagli, “Network intrusion detection system using deep learning,” Procedia Computer Science, vol. 185, pp. 239-247, 2021.
[29] G. Thamilarasu, A. Odesile, and A. Hoang, “An intrusion detection system for internet of medical things,” IEEE Access, vol. 8, pp. 181560-181576, 2020.
[30] I. Dutt, S. Borah and I. K. Maitra, “Immune system based intrusion detection system (IS-IDS): a proposed model,” IEEE Access, vol. 8, pp. 34929-34941, 2020.
[31] S. M. Kasongo, and Y. Sun, “A deep learning method with wrapper based feature extraction for wireless intrusion detection system,” Computers & Security, vol. 92, 101752, 2020.
[32] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani. “CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment,” Sensors, vol. 23, no. 13, 2023.
[33] D. Dolev and A. Yao, “On the security of public key protocols,” IEEE Transactions on Information Theory, vol. 29(2), pp. 198-208, 1983.
[34] R. Canetti and H. Krawczyk, “Universally composable notions of key exchange and secure channels,” Advances in Cryptology — EUROCRYPT, Springer Berlin Heidelberg, pp. 337–351, Amsterdam, The Netherlands, 2002.

[35] M. Wazid, A. K. Das, V. Bhat K, and A. V. Vasilakos, “LAM-CIoT: lightweight authentication mechanism in cloud-based IoT environment,” Journal of Network and Computer Applications, vol. 150, pp.102496, 2020.
[36] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, “Cloud centric authentication for wearable healthcare monitoring system,” IEEE Transactions on Dependable and Secure Computing, vol. 17(5), pp. 942-956, 2020.
[37] H. Nguyen, Q. Ngo, and V. Le, “IoT botnet detection approach based on PSI graph and DGCNN classifier,” IEEE International Conference on Information Communication and Signal Processing (ICICSP), Singapore, Singapore, 2018, pp. 118–122.
[38] J. Su, V. Danilo Vasconcellos, S. Prasad, S. Daniele, Y. Feng, and K. Sakurai, “Lightweight classification of IoT malware based on image recognition,” 42nd IEEE Annual Computer Software and Applications Conference (COMPSAC), vol. 02, Tokyo, Japan, 2018, pp. 664–669.
[39] S. Abbas, I. Bouazzi, S. Ojo, A. Al Hejaili, G. A. Sampedro, A. Almadhor, M. Gregus. “Evaluating deep learning variants for cyber- attacks detection and multi-class classification in IoT networks,” PeerJ Computer Science. 2024, 10:e1793.
[40] A. I. Jony, A. K. Arnob. “A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset,” Journal of Edge Computing, vol. 2024, pp. 1–15, 2024.
[41] E. N. Chinaechetam, C. I. Nwakanma, J. M. Lee, D. S. Kim. “Detecting cyberthreats in metaverse learning platforms using an explainable DNN,” Internet of Things, vol. 25, pp. 101046, 2024.
[42] S. Abbas, A. Al Hejaili, G. A. Sampedro, M. Abisado, A. Almadhor, T. Shahzad, K. Ouahada. “A novel federated edge learning approach for detecting cyberattacks in IoT infrastructures,” IEEE Access. vol. 11, pp. 112189-112198, 2023.
[43] T. T. H. Le, R. Wardhani, D. Putranto, U. Jo, and H. Kim. “Toward enhanced attack detection and explanation in intrusion detection system based IoT environment data,” IEEE Access, vol. 11, pp. 131661-131676, 2023.
[44] P. L. S. Jayalaxmi, R. Saha, G. Kumar, M. Conti and T. -H. Kim. “Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey,” IEEE Access, vol. 10, pp. 121173-121192, 2022.
[45] E. Benkhelifa, T. Welsh and W. Hamouda. “A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems,” IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3496-3509, 2018.
[46] S. Arisdakessian, O. A. Wahab, A. Mourad, H. Otrok and M. Guizani. “A Survey on IoT Intrusion Detection: Federated Learning, Game Theory, Social Psychology, and Explainable AI as Future Directions,” IEEE Internet of Things Journal, vol. 10, no. 5, pp. 4059-4092, 2023.
[47] Devkishen Sisodia, Jun Li, Samuel Mergendahl, and Hasan Cam. “A Two-Mode, Adaptive Security Framework for Smart Home Security Applications,” ACM Transactions Internet Things, vol. 5, no. 2, pp. 1-31, 2024.
[48] Simsek, I. “Authentication, Authorization, Access Control, and Key Exchange in Internet of Things,” ACM Transactions Internet Things, vol. 5, no. 2, pp. 1-30, 2024.
[49] Wang, H., Eklund, D., Oprea, A. & Raza, S. “FL4IoT: IoT Device Fingerprinting and Identification Using Federated Learning,” ACM Transactions Internet Things, vol. 4, no. 3, pp. 1-24, 2023.
[50] Attkan, A., Ranga, V. & Ahlawat. “P. A Rubik’s Cube Cryptosystem- based Authentication and Session Key Generation Model Driven in Blockchain Environment for IoT Security,” ACM Transactions Internet Things, vol. 4, no. 2, pp. 1-39, 2023.