***Journal of Artificial Intelligence and Technology*, 2026, 6, 129-150**
https://doi.org/10.37965/jait.2025.0880

RESEARCH ARTICLE

The Development of a Data Protection System in Healthcare Using Deep Learning Models

Vladislav Karyukin,[1,2] Olga Ussatova,[1,3] Aidana Zhumabekova,[1,2] Eric T. Matson,[4] Kuanysh Zhumabekova,[5] Nikita Ussatov,[1] and Yenlik Begimbayeva[1,3]

[1]Institute of Information and Computational Technologies, Almaty, Kazakhstan
[2]Al-Farabi Kazakh National University, Almaty, Kazakhstan
[3]G.Daukeev Almaty University of Energy and Communications, Almaty, Kazakhstan
[4]Purdue University, West Lafayette, USA
[5]Almaty Academy of the Ministry of Internal Affairs named M. Esbulatov, Almaty, Kazakhstan

(Received 24 July 2025; Revised 13 October 2025; Accepted 12 November 2025; Published online 02 December 2025)

Abstract: This research addresses the critical challenge of cybersecurity in healthcare by evaluating the effectiveness of machine learning (ML) and deep learning (DL) models in identifying and mitigating five significant cybersecurity threats: distributed denial-of-service (DDoS), man-in-the-middle (MitM), malware, phishing, and SQL injection. The study integrates a secure hardware–software architecture utilizing WireGuard, a lightweight, modern VPN protocol that establishes encrypted tunnels between network nodes, ensuring robust data integrity, confidentiality, and authenticated communication. Two ML models, support vector machine and random forest, and four DL architectures, dense neural networks (DNNs), convolutional neural networks-long short-term memory (CNN-LSTM), and LSTM-gated recurrent unit (LSTM-GRU), are systematically trained and tested using publicly available datasets specific to each threat category. The experimental outcomes demonstrate exceptional detection capabilities for structured network threats, with DNN and CNN-LSTM achieving accuracies and F1-scores from 95% to 97.6% for DDoS and MitM threats. In malware classification, the performance of DNN and CNN maintains precision and recall above 94%. Phishing and SQL injection attacks have lower classification scores of around 82% for most models. Visual analytics, including accuracy, loss plots, and confusion matrices, provide valuable insights into the convergence behaviors and sensitivity of different architectures, highlighting the strong generalization of DNN and variability in recurrent models. Overall, this research highlights the substantial potential of DL, combined with secure communication technologies like WireGuard, in enhancing healthcare cybersecurity, while also identifying areas for further development and optimization.

Keywords: cyber threats; DDoS; deep learning; Healthcare cybersecurity; malware; MitM; phishing; SQL injection; WireGuard

I. INTRODUCTION

Today, digital innovations [1] are changing healthcare by improving care quality, handling big data [2], streamlining documentation workflows [3], and intensifying security challenges [4] as systems and data scale. More frequent, sophisticated cyber threats [5] endanger the confidentiality and availability of healthcare information.

Healthcare organizations handle extensive volumes of sensitive data, such as personal information, medical histories, and diagnostic results, which makes them attractive targets for cybercriminals. Data leaks of confidential information can carry serious consequences for organizations.

Healthcare facilities manage vast amounts of confidential information, including personal details, medical histories, and diagnostic records, making them prime targets for cybercriminals. Data breaches and unauthorized access to medical or financial records can lead to severe consequences, including service disruptions, economic loss, and compromised patient safety.

Among the most critical cybersecurity threats in this domain are distributed denial-of-service (DDoS) [6], man-in-the-middle (MitM) [7], malware [8], phishing [9], and SQL injections [10]. DDoS attacks overwhelm healthcare networks with illegitimate traffic, rendering systems inaccessible and delaying critical procedures. For instance, a 2019 DDoS attack on a US healthcare provider caused widespread outages, disrupting patient appointments and communications.

MitM attacks intercept communications between healthcare professionals and medical devices, enabling unauthorized access to sensitive information [11]. For example, attackers may capture data transmitted from medical equipment to electronic health record (EHR) systems, leading to data breaches or manipulation. Malware, such as ransomware, encrypts critical data and demands payment for access restoration [12]. A notable case is the 2017 WannaCry attack, which disrupted healthcare services globally and endangered patient safety.

Phishing attacks deceive healthcare staff via fraudulent emails or messages, tricking them into revealing credentials or clicking malicious links [13]. For instance, a phishing email impersonating a medical supplier could prompt staff to share login information, exposing the network to further threats. SQL injection exploits application vulnerabilities by injecting malicious SQL code,

Corresponding author: Olga Ussatova (e-mail: olgaussatova@gmail.com).

allowing attackers to access or alter databases. A poorly secured hospital website, for example, could be exploited to extract confidential patient records [14].

Given the severity and evolving nature of these threats, healthcare institutions must adopt advanced tools capable of detecting and preventing real-time cyberattacks. The traditional methods are mostly not effective against complex types of attacks and threats. Therefore, machine learning (ML) [15] and deep learning (DL) [16] approaches have become strong cybersecurity tools in healthcare environments.

These technologies offer several key benefits:

– **Detecting threats** and initiating immediate countermeasures [17].

– **Adaptability to emerging threats**: Continuously learning and evolving to recognize novel attack patterns [18].

– **Automation and reduced human error:** Minimizing reliance on human intervention and decreasing error rates [19].

– **Scalability**: Effectively handling and securing large volumes of sensitive patient data [20].

– **Detection of hidden threats:** Uncovering subtle or previously undetected threats that traditional systems might miss [21].

As cyber threats grow in complexity, integrating ML and DL models into healthcare cybersecurity frameworks becomes an essential part of the systems.

In addition to software-based defenses, robust integration of both hardware and software is crucial for establishing a secure healthcare IT infrastructure. Therefore, a comprehensive hardware–software architecture was developed to ensure secure authentication, establish encrypted communication channels, and integrate flow encryption keys directly into hardware components. This system addresses several critical security challenges, including the protection of communication channels through cryptographic methods, secure information transfer within isolated networks, effective routing of network participants, and the secure transmission of cryptographic keys. The routing of cryptographic keys was implemented using WireGuard, a modern VPN protocol known for its simplicity and high performance. WireGuard establishes tunnel interfaces (e.g., *wg0* and *wg1*) on top of existing network interfaces, which are configured using the ifconfig, ip, and route tools. Tunnel behavior is controlled using the *wg* tool, which associates IP addresses with public keys and remote endpoints to manage communication between nodes securely. This configuration provides a lightweight and secure foundation for encrypted communication in healthcare systems.

In this study, we combine these architectural and cryptographic advancements with ML and DL models for detecting cyber threats in healthcare. Among ML models, the support vector machine (SVM) and random forest (RF) are taken for the experiments, along with a convolutional neural network (CNN), a dense neural network (DNN), a hybrid CNN-long short-term memory (CNN-LSTM) network, and an LSTM-gated recurrent unit (LSTM-GRU) network. Other parts of the paper include a related methodology, an implemented approach, a description of experimental results, and a conclusion of the whole research work.

II. RELATED WORKS

The presented cyber threats are covered in many different research works. The paper [22] introduces an advanced intrusion detection system (IDS) by proposing an improved variant of the LSTM model designed to detect DDoS attacks with the use of the model which integrates convolutional, bidirectional LSTM (Bi-LSTM), and bidirectional gated recurrent unit (Bi-GRU) layers, allowing it to effectively capture both temporal and spatial features from Internet of Things (IoT) network traffic. The results demonstrate exceptional accuracy and F1-scores of 0.95 and an Area Under the Curve Receiver Operating Characteristic (AUC-ROC) score of 0.99.

The main research results of the paper [23] indicate that MitM attacks can be effectively detected with the use of LSTM, SVM, and RF models. The experiments demonstrate that RF has the highest accuracy score of 0.94, while LSTM is also close, reaching a score of 0.92. Only the SVM model's accuracy score is below 0.90, achieving a score of 0.86. The results demonstrate enhanced robustness in handling noisy or incomplete data, as well as strong generalization capabilities to unseen scenarios. Furthermore, comparative analysis confirms that the proposed models improved predictive quality, making them suitable for practical, real-world applications.

The main research results of the paper [24] indicate the implementation of Naïve Bayes (NB), k-nearest neighbors (KNN), and CNN in the detection of MitM attacks in the IoT environment. The comparison of models confirms a reduction of computational complexity while maintaining the high predictive quality. NB, KNN, and CNN achieve accuracy scores of 0.94, 0.97, and 0.99, respectively.

The study [25] presents comprehensive research on developing a novel DL AutoEncoder model with XGBoost for detecting and preventing MitM in IoT networks. The research thoroughly describes the limitations of traditional IDSs, which often struggle with high-dimensional data, evolving attack strategies, and real-time constraints. Applied to the intrusion detection dataset, the model achieved accuracy, precision, recall, and F1-scores of 0.97, 0.96, 0.95, and 0.96, outperforming the standard models, such as RF, SVM, and standard XGBoost. The approach also demonstrated a superior AUC-ROC value of 0.97, indicating robust discrimination between benign and malicious traffic.

The study [26] describes the TuneDroid technique for detecting Android malware. It explicitly addresses three prevailing challenges: obfuscation, accuracy detection, and computational efficiency in the presented domain. The model was evaluated on a dataset with 3000 benign and 3000 malicious Android applications. TuneDroid provided an opportunity to achieve an accuracy score of 0.99. This approach demonstrates significant improvements over traditional static analysis methods, highlighting the potential of dynamic tuning in malware detection.

The research in the paper [27] focuses on the rising threat of phishing websites by leveraging ML and DL models within cloud and fog computing contexts. The study introduces a new dataset containing benign and malicious traffic and employs RF and SVM models. The SVM model reached an accuracy score of 98%.

The research [28] presents a comprehensive evaluation of phishing email detection models with 14 ML and DL algorithms across 10 datasets, including a newly created merged dataset. The datasets range from classic corpora like Enron and SpamAssassin to specialized sets such as Nazario and Nigerian scams, ensuring diversity in phishing patterns. For ML models, Term Frequency – Inverse Document Frequency (TF-IDF) vectorization and preprocessing were applied, while DL models were trained on raw text to capture semantic nuances. The results showed that BERT and RoBERTa transformer models achieved the highest scores of 0.98 and 0.99, respectively. These models maintained a strong precision–recall balance, crucial for reducing both false positives and false negatives. In contrast, graph convolutional networks performed poorly because email text is linear. Among ML models, the SGD Classifier delivered the best overall results with 0.98 average accuracy, showing adaptability across datasets. The study proved the transformer-based models to be the most effective for phishing detection, enhancing resilience against evolving phishing attacks.

The paper [29] analyzes the rapid increase in web application attacks, both in frequency and complexity. The proliferation of these attacks is primarily fueled by the extensive data available online, which has become highly attractive to cybercriminals. Structured Query Language Injection (SQLi) attacks are especially prevalent and pose severe risks to the information in various databases. In this paper, the classification of SQLi was implemented with the LSTM model. It was tested across three datasets and got accuracy scores from 0.98 to 0.99.

The research works [30–33] implemented two innovative CNN models: SQL Injection-attack Detection Network-1 (SIDNet-1) and SQL Injection-attack Detection Network-2 (SIDNet-2). These models are specifically designed to classify and detect SQLi, thereby enhancing the security of web applications. In the classification results, SIDNet-1 and SIDNet-2 achieved remarkable accuracy scores of 0.97 and 0.98, respectively, on the SQLIV2.

Beyond the evaluated threat models, a broader and often under-examined concern was the adversarial vulnerability gap, referring to the discrepancy between threats considered during evaluation and the full spectrum of potential adversarial manipulations that neural networks may face in practice. While the neighborhood expectation attribution attack (NEAA) demonstrates strong transferability and effectively disrupts intrinsic feature representations across models, an inherent limitation remains in measuring robustness solely against tested attacks. This gap highlights a critical risk that is defined by defenses and robustness strategies, which may appear effective under restricted or predefined attack settings but remain susceptible to unforeseen adversarial strategies under different assumptions [34].

Another case is related to the Multi-Feature Attention Attack (MFAA), which demonstrates effectiveness in improving transferability through multilayer feature fusion and ensemble attention mechanisms. However, it implicitly exposes a structural weakness in deep neural networks: their reliance on overlapping, model-agnostic feature hierarchies that can be systematically manipulated. By exploiting cross-layer semantic consistency and disrupting shared category-related representations, MFAA reveals that even advanced models with defensive strategies remain susceptible when adversarial signals align with naturally learned semantics. This highlights a fundamental concern: robustness evaluations that focus only on known threat models or architectures may underestimate real-world susceptibility to novel, feature-aware adversarial strategies [35].

III. METHODOLOGY

This research employs an ML- and DL-based approach to classify five major cybersecurity threats: DDoS, MitM, malware, phishing, and SQL injection. The methodology includes a list of phases: dataset collection, data preprocessing and normalization, feature selection/extraction, model training, and performance evaluation. All these steps allow the support of the consistency and heterogeneity of datasets.

While this work primarily focuses on DL-based threat detection, secure data handling is crucial during both the training and deployment stages in sensitive environments, such as healthcare. Hardware–**software architecture** is implemented to support **secure communication channels and facilitate the routing of cryptographic keys**. The system leverages **WireGuard**, a lightweight and modern VPN protocol, to establish encrypted tunnels between network nodes. These secure interfaces (e.g., *wg0* and *wg1*) operate on top of conventional interfaces and are configured via standard tools such as ifconfig, ip, and route. WireGuard associates each peer with a public key and an internal IP address, allowing only authenticated devices to exchange data. The *wg* utility facilitates real-time configuration and monitoring of the encrypted channels. This infrastructure complements the ML and DL models by safeguarding **data integrity, confidentiality, and compliance** during experimentation and operational deployment. This setup provides a lightweight, scalable, and cryptographically sound communication framework that complements the DL models by ensuring data integrity and confidentiality during both model training and real-time operation. The entire framework is illustrated in Fig. 1.

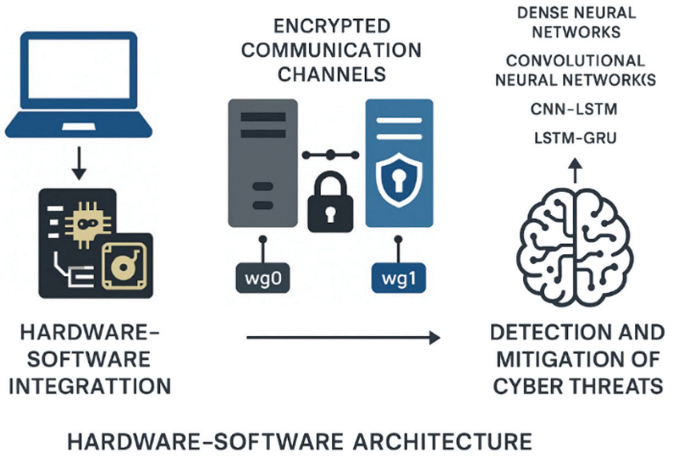

ENCRYPTED COMMUNICATION CHANNELS
DENSE NEURAL NETWORKS
CONVOLUTIONAL NEURAL NETWORK(S
CNN-LSTM
LSTM-GRU
wg0
wg1
HARDWARE-SOFTWARE INTEGRATTION
DETECTION AND MITIGATION OF CYBER THREATS
HARDWARE-SOFTWARE ARCHITECTURE

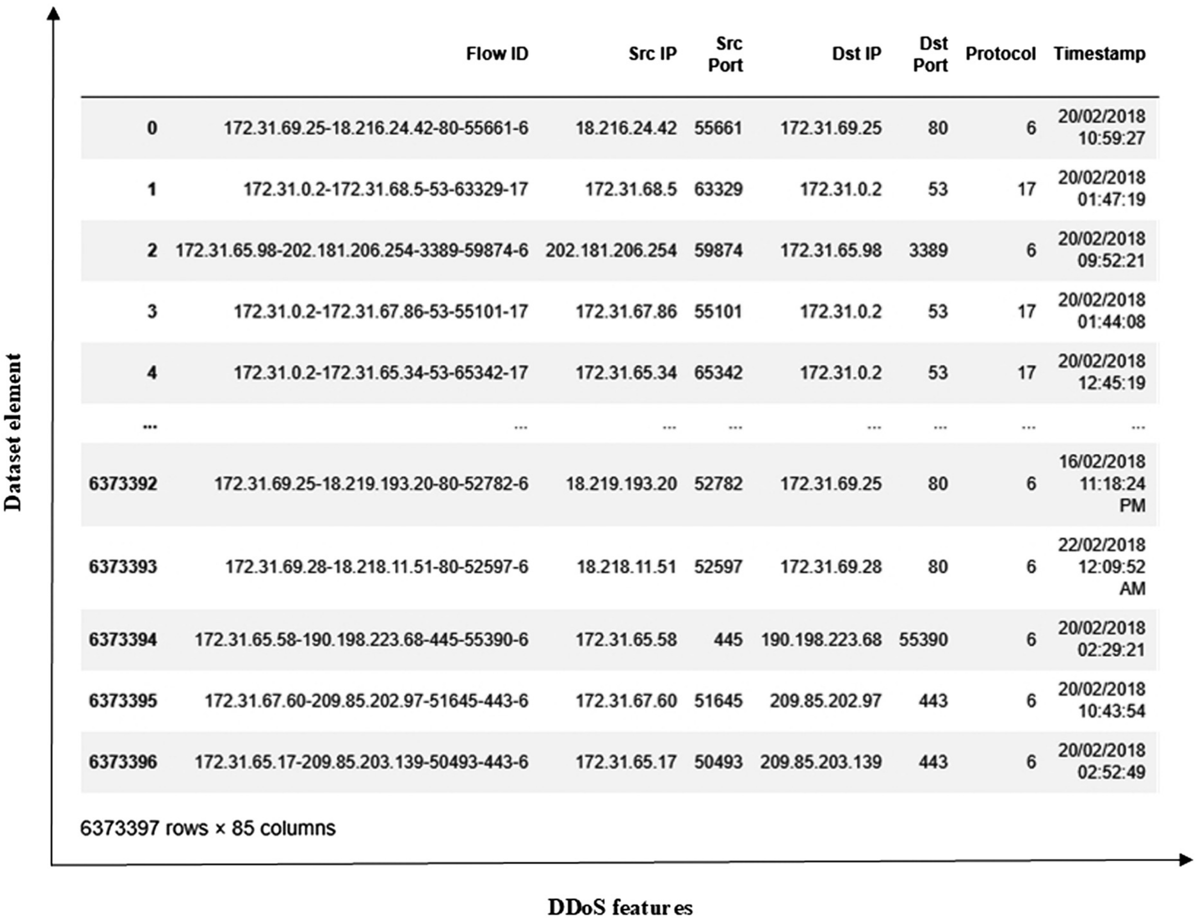
Fig. 1. The system's infrastructure.

This secure architecture provides an opportunity for protected data transmission; however, subsequent steps focus on deploying a protection system against cyber threats using deep learning models. The steps toward the preparation of these models are described in the next subsections.

A. DATASETS

To train efficient DL models for detecting DDoS, MitM, malware, phishing, and SQL injection cyber threats, five advanced datasets are compiled for each of these categories. These datasets are collected from different publicly available sources.

The DDoS dataset (https://www.kaggle.com/code/taruntambrahalli/ddos-nbc) is a comprehensive collection of various attacks compiled from CICDoS2016, CICIDS2017, and CICIDS2018 sources. These datasets are created in different years using diverse DDoS traffic generation tools. The extracted DDoS flows are then combined with benign traffic, which is separately extracted from the same original datasets, to create a single, unified dataset. The features of the dataset contain comprehensive information about IP addresses, ports, sizes of data packets, etc. The whole dataset includes 6373397 elements and 84 features. The whole DDoS dataset is shown in Fig. 2.



	Flow ID	Src IP	Src Port	Dst IP	Dst Port	Protocol	Timestamp
0	172.31.69.25-18.216.24.42-80-55661-6	18.216.24.42	55661	172.31.69.25	80	6	20/02/2018 10:59:27
1	172.31.0.2-172.31.68.5-53-63329-17	172.31.68.5	63329	172.31.0.2	53	17	20/02/2018 01:47:19
2	172.31.65.98-202.181.206.254-3389-59874-6	202.181.206.254	59874	172.31.65.98	3389	6	20/02/2018 09:52:21
3	172.31.0.2-172.31.67.86-53-55101-17	172.31.67.86	55101	172.31.0.2	53	17	20/02/2018 01:44:08
4	172.31.0.2-172.31.65.34-53-65342-17	172.31.65.34	65342	172.31.0.2	53	17	20/02/2018 12:45:19
...	...	...	...	...	...	...	...
6373392	172.31.69.25-18.219.193.20-80-52782-6	18.219.193.20	52782	172.31.69.25	80	6	16/02/2018 11:18:24 PM
6373393	172.31.69.28-18.218.11.51-80-52597-6	18.218.11.51	52597	172.31.69.28	80	6	22/02/2018 12:09:52 AM
6373394	172.31.65.58-190.198.223.68-445-55390-6	172.31.65.58	445	190.198.223.68	55390	6	20/02/2018 02:29:21
6373395	172.31.67.60-209.85.202.97-51645-443-6	172.31.67.60	51645	209.85.202.97	443	6	20/02/2018 10:43:54
6373396	172.31.65.17-209.85.203.139-50493-443-6	172.31.65.17	50493	209.85.203.139	443	6	20/02/2018 02:52:49

6373397 rows × 85 columns

Fig. 2. The DDoS dataset.

The MitM dataset (https://www.kaggle.com/datasets/ymirsky/network-attack-dataset-kitsune/data) is also a complex collection of network attack datasets that have been captured from network IoT devices. It has various types of attacks, such as Active Wiretap, OS Scan, SSDP Flood, Mirai Botnet, and others. The entire dataset comprises both benign and malicious traffic, enabling the development of models that can distinguish between normal and attack behaviors. Each network packet is represented by features obtained by the Kitsune Network Intrusion Detection System (NIDS), which provides a more detailed analysis and model training. The MitM dataset comprises 2504267 elements and 115 features, as shown in Fig. 3.

The malware dataset (https://github.com/saurabh48782/Malware_Classification/blob/master/MalwareData.csv) is designed for binary classification of malicious and benign objects. This dataset is adaptable for different DL algorithms for malware detection. Malicious software poses significant threats to information systems. There is a large variety of malware, such as worms, viruses, trojans, and others. The entire dataset comprises 216351 malware elements and 53 features. It is shown in Fig. 4.

The phishing dataset (https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls) is a substantial resource for developing and testing ML and DL models to detect phishing websites. It comprises approximately 549346 elements, each consisting of a URL and a corresponding label indicating whether the site is legitimate or a phishing attempt. The dataset's data structure includes a URL (the web address to be analyzed) and a Label (the URL classification, where 0 denotes a legitimate site and 1 indicates a phishing site). The dataset is shown in Fig. 5.

The SQLi dataset (https://www.kaggle.com/datasets/gambleryu/biggest-sql-injection-dataset) is designed to support research and development in the detection of SQL attacks. The corresponding dataset is effectively used to train advanced DL models aimed at identifying SQLi patterns. This dataset includes the Query and Label columns, consisting of 148326 elements. The dataset is shown in Fig. 6.

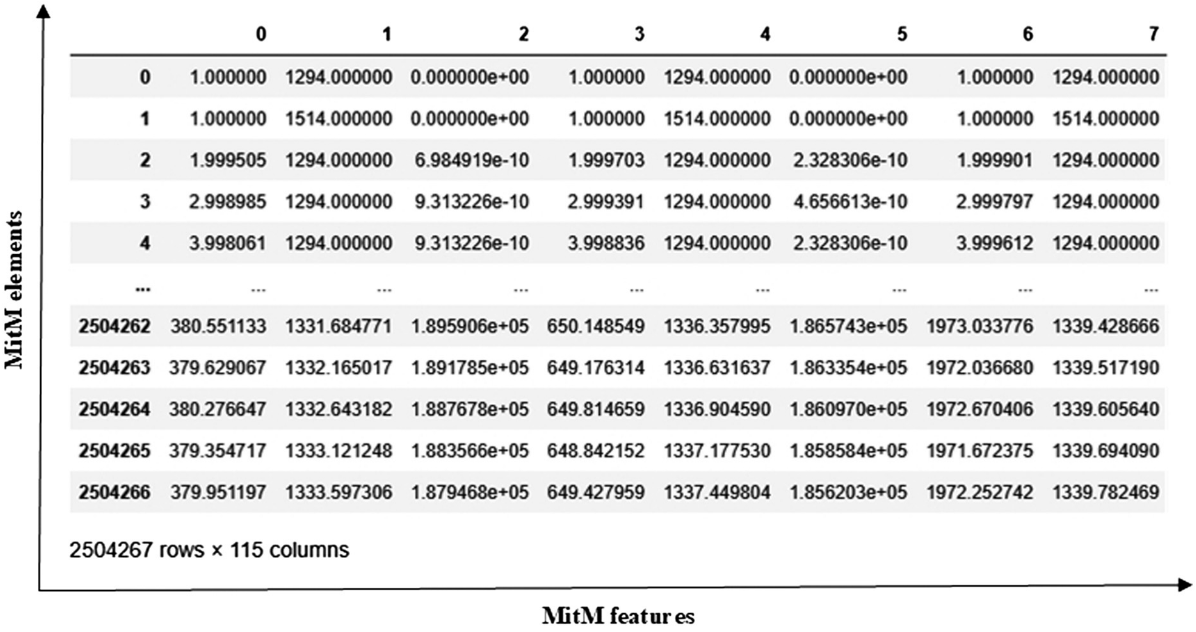
B. DATA PROCESSING AND NORMALIZATION

To ensure consistent and effective model performance, distinct preprocessing pipelines were implemented based on the nature of each dataset. For the DDoS, MitM, and malware datasets, which typically consist of structured network traffic data, the preprocessing began with data cleaning to remove duplicate records, incomplete entries, or anomalies.

Min–max normalization is a feature-scaling technique for transforming feature values into a scale from 0 to 1. It ensures all features contribute equally to algorithms. The min–max technique is computed by (1):

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{1}$$

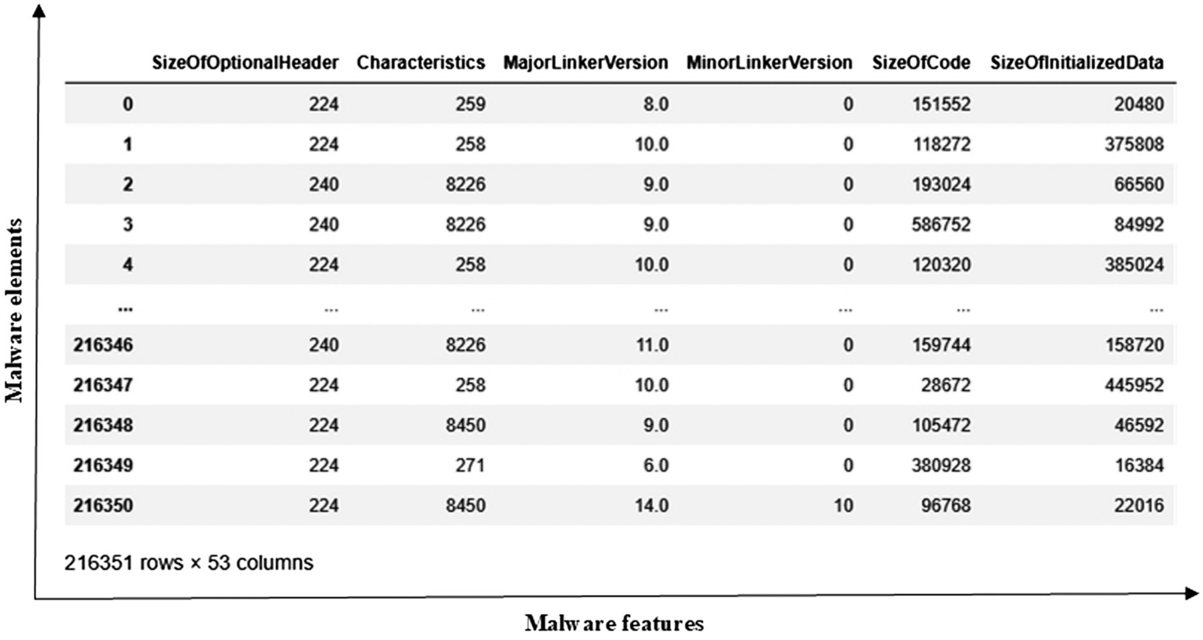
where x is an initial value, x_{min} is a minimum value, x_{max} is a maximum value, and x' is a normalized value.



	0	1	2	3	4	5	6	7
0	1.000000	1294.000000	0.000000e+00	1.000000	1294.000000	0.000000e+00	1.000000	1294.000000
1	1.000000	1514.000000	0.000000e+00	1.000000	1514.000000	0.000000e+00	1.000000	1514.000000
2	1.999505	1294.000000	6.984919e-10	1.999703	1294.000000	2.328306e-10	1.999901	1294.000000
3	2.998985	1294.000000	9.313226e-10	2.999391	1294.000000	4.656613e-10	2.999797	1294.000000
4	3.998061	1294.000000	9.313226e-10	3.998836	1294.000000	2.328306e-10	3.999612	1294.000000
...	...	...	...	...	...	...	...	...
2504262	380.551133	1331.684771	1.895906e+05	650.148549	1336.357995	1.865743e+05	1973.033776	1339.428666
2504263	379.629067	1332.165017	1.891785e+05	649.176314	1336.631637	1.863354e+05	1972.036680	1339.517190
2504264	380.276647	1332.643182	1.887678e+05	649.814659	1336.904590	1.860970e+05	1972.670406	1339.605640
2504265	379.354717	1333.121248	1.883566e+05	648.842152	1337.177530	1.858584e+05	1971.672375	1339.694090
2504266	379.951197	1333.597306	1.879468e+05	649.427959	1337.449804	1.856203e+05	1972.252742	1339.782469

2504267 rows × 115 columns

Fig. 3. The MitM dataset.



	SizeOfOptionalHeader	Characteristics	MajorLinkerVersion	MinorLinkerVersion	SizeOfCode	SizeOfInitializedData
0	224	259	8.0	0	151552	20480
1	224	258	10.0	0	118272	375808
2	240	8226	9.0	0	193024	66560
3	240	8226	9.0	0	586752	84992
4	224	258	10.0	0	120320	385024
...	...	...	...	...	...	...
216346	240	8226	11.0	0	159744	158720
216347	224	258	10.0	0	28672	445952
216348	224	8450	9.0	0	105472	46592
216349	224	271	6.0	0	380928	16384
216350	224	8450	14.0	10	96768	22016

216351 rows × 53 columns

Fig. 4. The malware dataset.

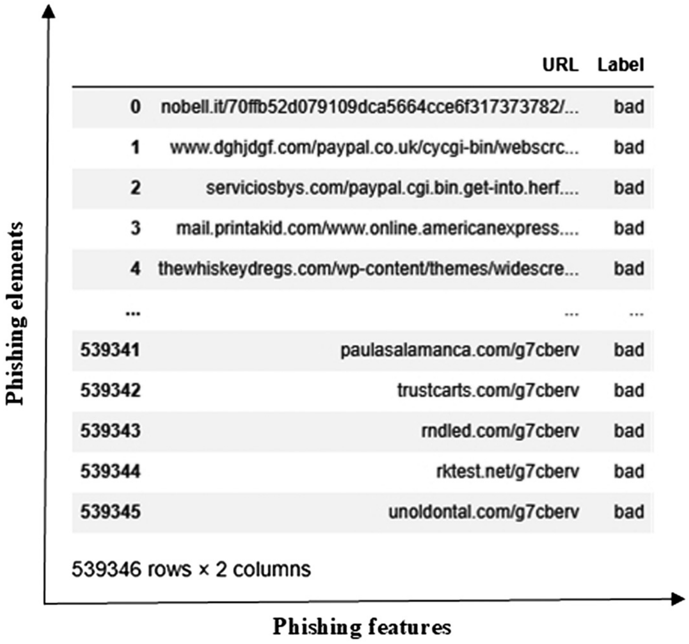
C. FEATURE SELECTION

After normalization, feature selection was performed using the chi-square metric. This technique gives the most important features. The chi-square metric measures the dependence between each feature and the target class label using the chi-square (χ^2) test of independence. This feature selection technique is especially suitable for discretized numerical features, helping to reduce dimensionality while preserving meaningful input. The chi-square metric is computed in the following way (2):

$$x_c^2 = \sum \frac{(N_i - M_i)^2}{M_i}, \qquad (2)$$

where N is an observed value, M is an expected value, and c is a degree of freedom.

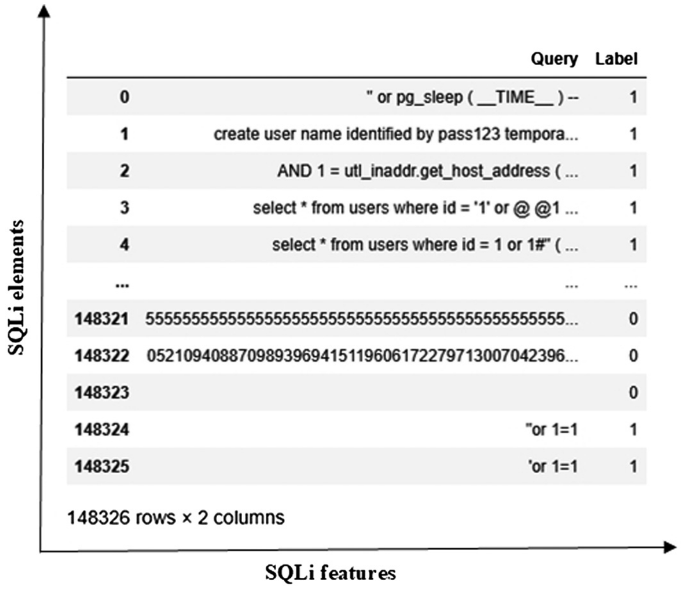
In this work, the 20 best features of DDoS, MitM, and malware datasets were chosen in the feature selection stage.



	URL	Label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	bad
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	bad
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	bad
3	mail.printakid.com/www.online.americanexpress....	bad
4	thewhiskeydregs.com/wp-content/themes/widescre...	bad
...	...	...
539341	paulasalamanca.com/g7cberv	bad
539342	trustcarts.com/g7cberv	bad
539343	rndled.com/g7cberv	bad
539344	rktest.net/g7cberv	bad
539345	unoldontal.com/g7cberv	bad

539346 rows × 2 columns

Fig. 5. The phishing dataset.



	Query	Label
0	" or pg_sleep (__TIME__) --	1
1	create user name identified by pass123 tempora...	1
2	AND 1 = utl_inaddr.get_host_address (...	1
3	select * from users where id = '1' or @ @1 ...	1
4	select * from users where id = 1 or 1#" (...	1
...	...	...
148321	555555555555555555555555555555555555555555555...	0
148322	0521094088709893969415119606172279713007042396...	0
148323		0
148324	"or 1=1	1
148325	'or 1=1	1

148326 rows × 2 columns

Fig. 6. The SQL injection dataset.

D. FEATURE EXTRACTION

In contrast to other datasets, the phishing and SQL injection datasets mainly contain raw text data, such as URLs and SQL query strings. The conversion of textual strings is implemented using the TF-IDF metric, which converts textual data into numerical vectors by evaluating their importance in the whole dataset. This representation effectively captures both common structures that may signal phishing or SQL injection attacks.

TF-IDF is computed by (3):

$$TF - IDF = TF \times IDF \tag{3}$$

The significance of t_i is evaluated as (4):

$$TF(t_i) = \frac{m_i}{\sum_{i=1}^{k} m_i}, \tag{4}$$

where m_i is the number of times a word t_i takes place in a sentence, and $\sum_{i=1}^{k} m_i$ is the total number of words in a sentence.

IDF is the inverse of the frequency of a word in a sentence and is computed as (5):

$$IDF(t_i,S) = \log \frac{|S|}{|(s_i \supset t_i)|}, \tag{5}$$

where $|S|$ is the full number of sentences; $|(s_i \supset t_i)|$ is the number of sentences where t_i takes place.

E. ML AND DL MODELS

The datasets in this work are classified with a bunch of ML and DL models.

An SVM is a type of algorithm efficiently used for classification and regression tasks. It focuses on finding the optimal decision boundary for separating data points of different classes with a margin. The hyperplane of SVM is defined as (6):

$$w \times x + b = 0, \tag{6}$$

where $w = (w_1, w_2, \ldots, w_n)$ is a weight vector, $x = (x_1, x_2, \ldots, x_n)$ is a data vector, and b is a bias.

The SVM model is shown in Fig. 7.

An RF is an ensemble learning method that builds a collection of decision trees and combines their predictions to improve accuracy and stability. It works by creating multiple random samples from the dataset through a process called bagging, where each sample is drawn with replacement and used to train an individual tree. To further diversify the trees, RF selects a random subset of features at each split, which reduces correlation among the trees. Once all the trees are built, their predictions are aggregated: for classification tasks, the model outputs the class chosen by the majority of trees. The RF model is shown in Fig. 8.

A DNN is a fully connected network and the basic DL architecture. Every neuron of the layer is connected to neurons of the next layer. It allows the network to combine information from all features, enabling it to capture complex, nonlinear relationships within the data. Due to its versatility, DNN is good for various classification assignments, representing the learning process.

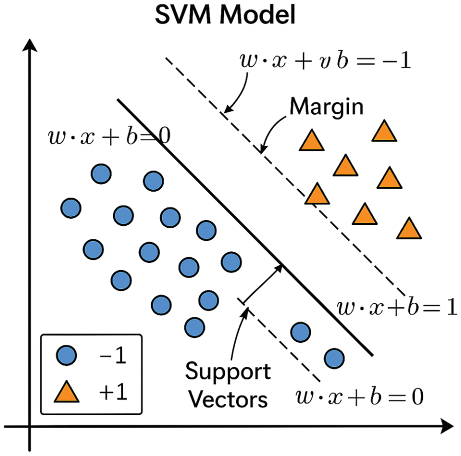


Fig. 7. The SVM model.

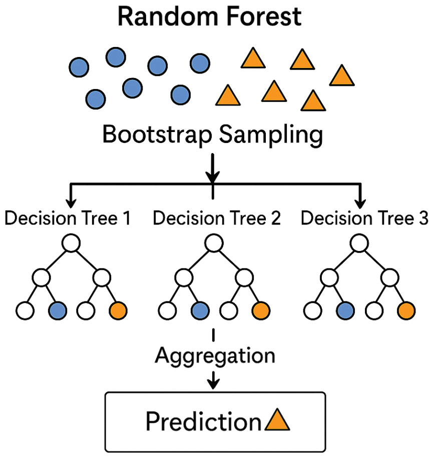


Fig. 8. The random forest model.

The structure of a DNN includes an input layer, which receives a fixed-size numerical vector. The input vector is followed by several hidden layers where weights are multiplied by neuron values and summed up with each other and a bias. The results of the multiplications then undergo Rectified Linear Unit (ReLU) or sigmoid activation functions. This computation is shown as (7):

$$s = f\left(\sum_{i=1}^{n} w_i \times x_i + b\right) \tag{7}$$

where x_i are the input values, w_i are the weights, b is the bias, and f is the activation function.

The output layer provides the final prediction, with the sigmoid function for binary classification. The DNN scheme is shown in Fig. 9.

A CNN is a DL architecture designed to learn spatial hierarchies of features from input data. It is especially effective for analyzing special structures, like images, but can also be applied to other datasets. Unlike the traditional DNNs with feedforward-connected layers, CNNs implement specialized layers to learn hierarchies of patterns through local connectivity and weights. The core components of CNN are convolutional layers, where filters slide across the input data. These convolutional layers are followed by ReLU activation functions, which introduce nonlinearity in models and facilitate the handling of complex functions. After applying ReLU functions, max pooling layers decrease the dimensionality of feature maps. Pooling also summarizes the most essential features in each region. The output is flattened into a one-dimensional vector when the dimensions are sufficiently reduced through successive convolution and pooling layers. The output layer of the network gives the final prediction of the sigmoid function. The mathematical computations are shown in (8)–(11):

1. Convolutional operations:

$$S(i,j) = (I * \mathrm{K})(i,j) = \sum_{u=0}^{m-1}\sum_{v=0}^{n-1} \mathrm{K}(u,v) I(i+u,\ j+v) \tag{8}$$

where I is an input image of size $H \times W$ and K is a kernel filter of size $m \times n$.

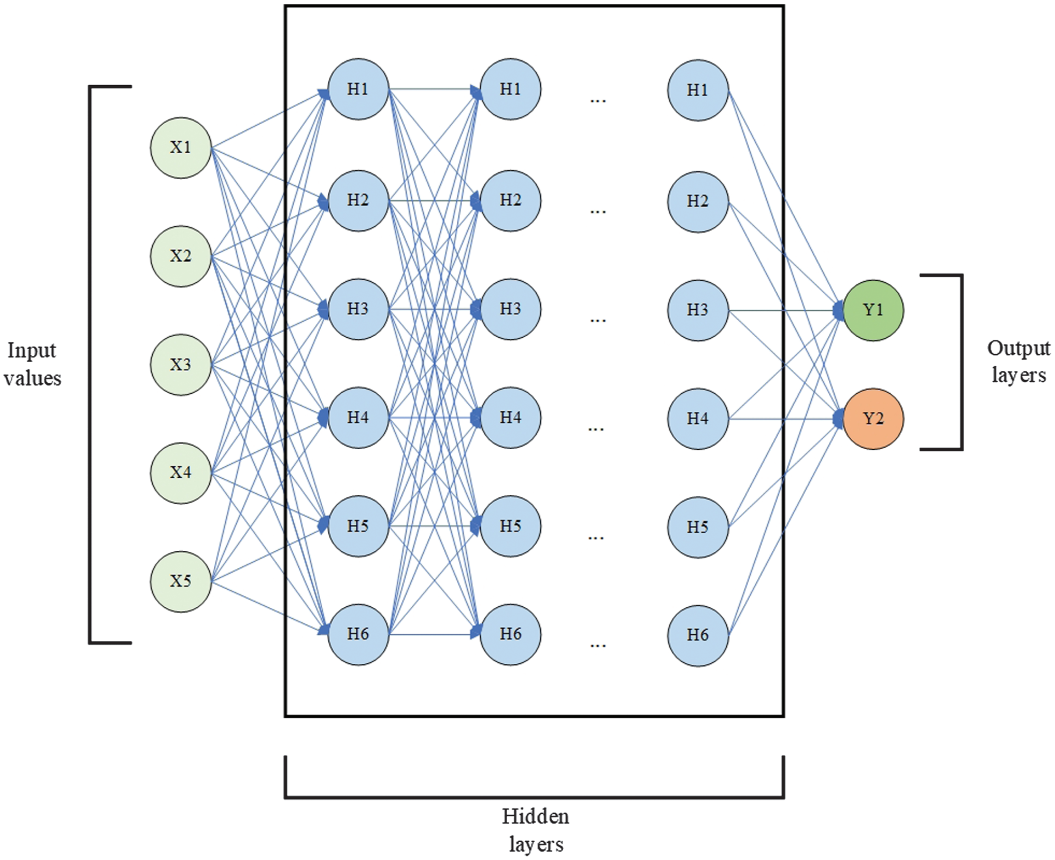


Fig. 9. Dense neural network.

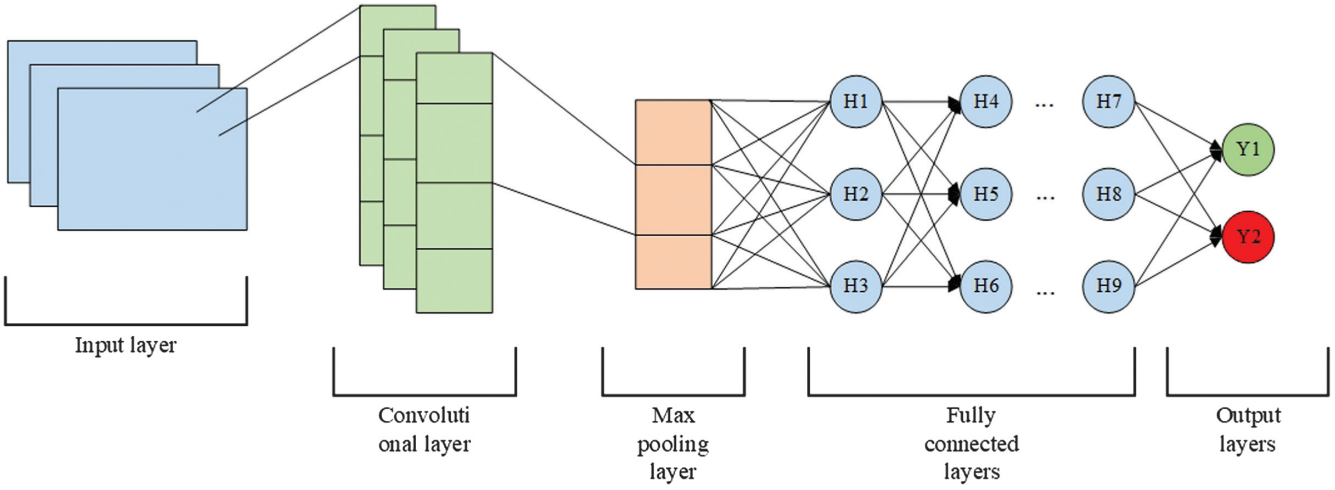


Fig. 10. Convolutional neural network.

2. The ReLU activation function:

$$ReLU(x) = \max(0,x) \tag{9}$$

3. Max pooling:

$$P(i,j) = \max_{0\le u<k,0\le v<k} I(i+u,j+v) \tag{10}$$

4. Fully connected layer:

$$y = f(Wx+b) \tag{11}$$

where W is a weight matrix, b is a bias vector, and f is an activation function.

The scheme of CNN is shown in Fig. 10.

A CNN-LSTM is a hybrid DL architecture that merges the strengths of CNN and LSTM networks. The model begins with a

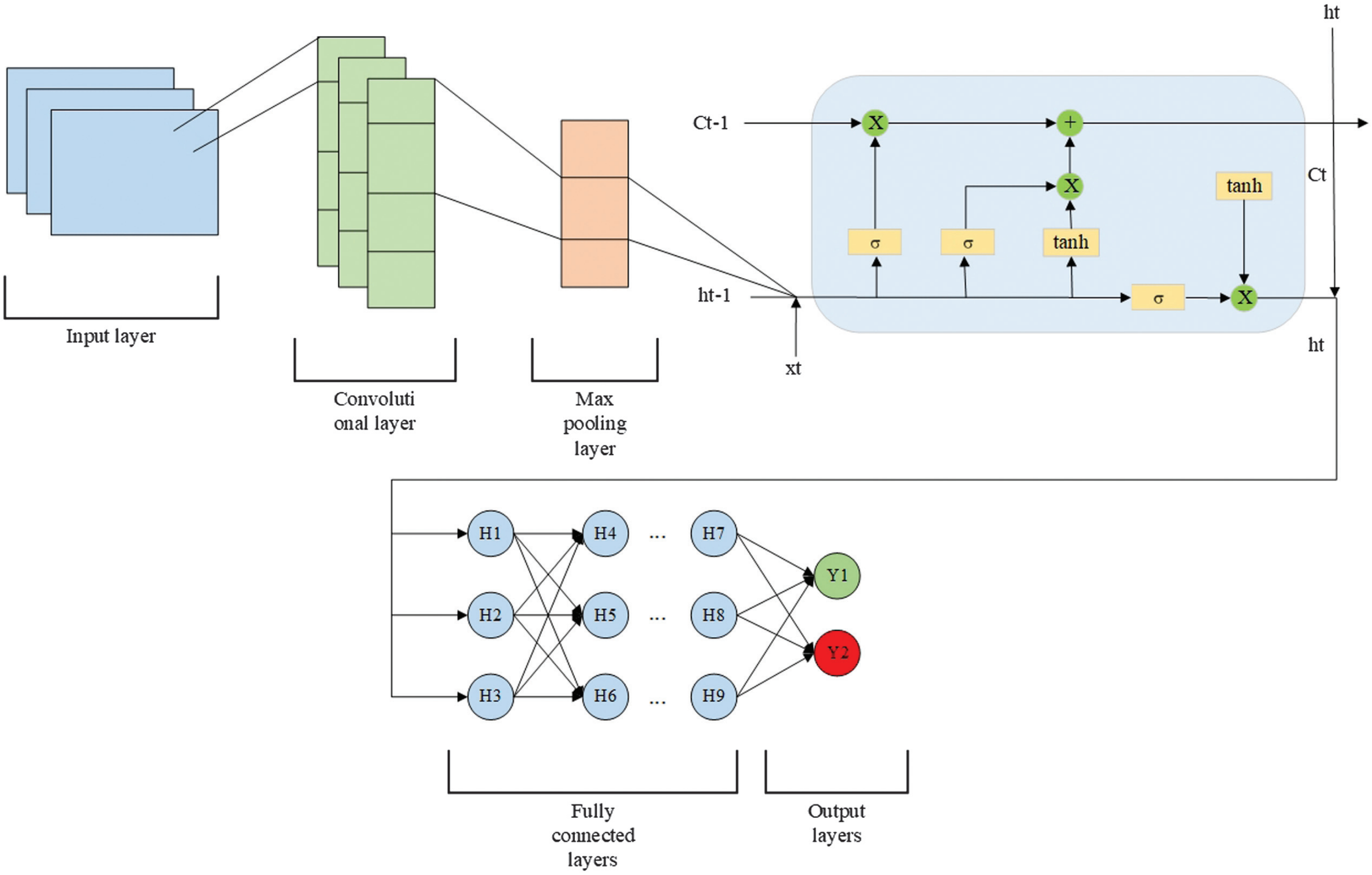


Fig. 11. CNN-LSTM.

convolutional layer, applying multiple filters to the input sequences, such as those with shapes of (20, 1) or (100, 1), depending on the dataset—to capture local spatial patterns. The ReLU activation function introduces nonlinearity, enabling the detection of complex feature interactions. This is followed by a global max pooling layer, which compresses each feature map into a single value, reducing dimensionality while preserving the most salient features. The pooled features are then passed through a fully connected dense layer, projecting them into a higher-dimensional space. Next, an LSTM layer processes these representations, leveraging its gated architecture to retain and model temporal dependencies in the data. Additional dense layers with ReLU activation further refine the learned features, transforming them into a task-specific representation. The model concludes with a single-neuron output layer using a sigmoid activation, producing a probability score in the range of 0 and 1, suited for binary classification.

The CNN-LSTM scheme is shown in Fig. 11.

An LSTM-GRU neural network is a hybrid recurrent architecture that combines two types of GRUs: LSTM and GRU. Both LSTM and GRU are designed to process long-term dependencies, but they do so in different ways. Combining them in a single network can leverage their complementary strengths for enhanced sequence modeling. The first layer of the combined model is an LSTM layer with the input data of (20, 1) or (100, 1) shapes, depending on the dataset type. The output from the LSTM is then put into a **GRU layer**. The GRU is simpler and more computationally efficient than LSTM and is used here to process the temporal patterns extracted by the LSTM further. Following the GRU, a standard **dense layer with a single unit** and a **sigmoid activation function** outputs a probability between 0 and 1. The scheme of LSTM-GRU is shown in Fig. 12.

The presented DL models are actively used in the subsequent experiments.

IV. EXPERIMENTAL RESULTS

Experimental results on classifying cyber threats were conducted using four DL models: DNN, CNN, CNN-LSTM, and LSTM-GRU. Each architecture was trained and tested on labeled threat data to evaluate its effectiveness in identifying malicious activity. The models' performance was estimated using accuracy, precision, recall, and F1-score classification metrics (12)–(15):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F1 - score = 2\frac{Rrecision \cdot Recall}{Rrecision + Recall} \tag{15}$$

where TP is true positives, TN is true negatives, FP is false positives, and FN is false negatives.

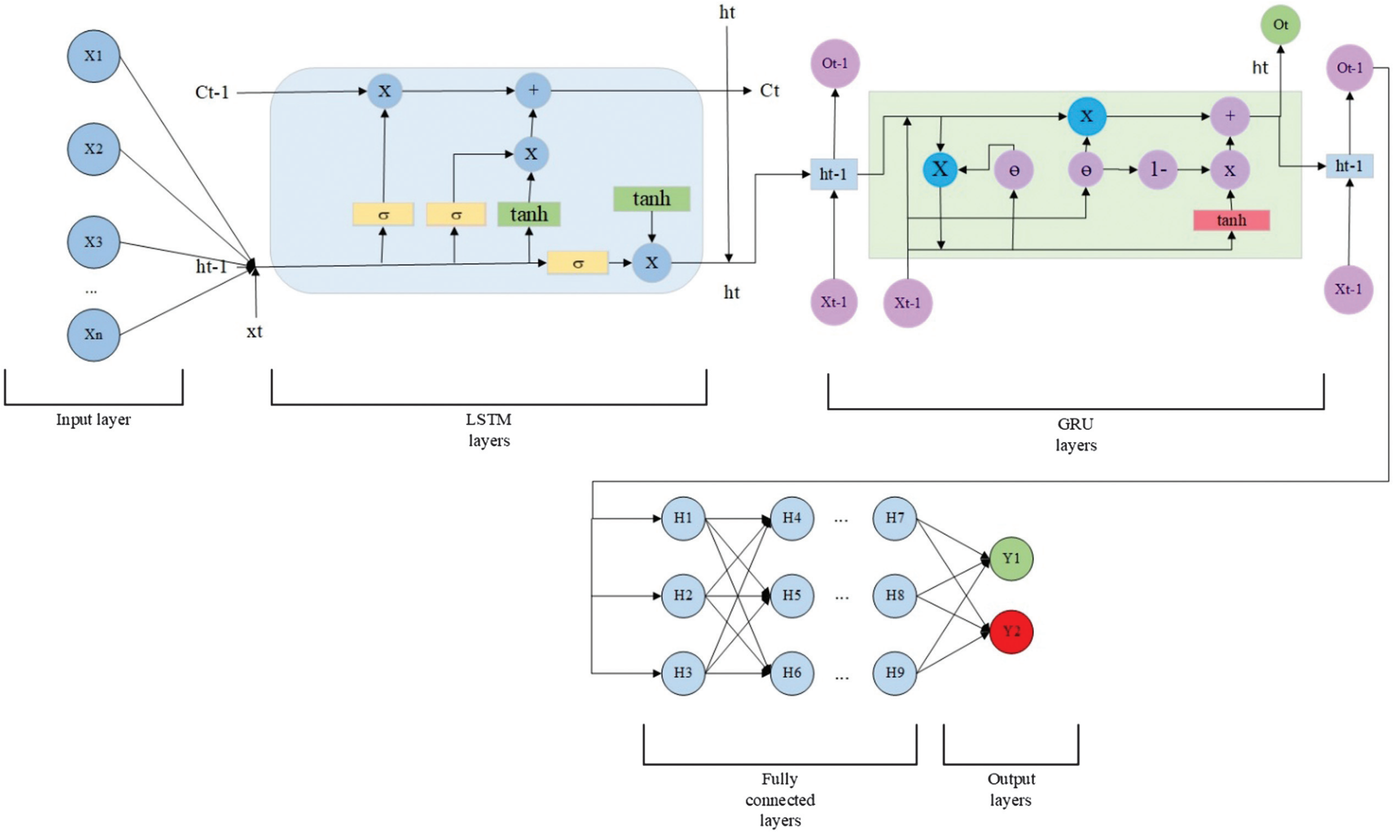


Fig. 12. LSTM-GRU.

Table I. Classification results of DDoS attacks

Metrics	SVM	Random forest	DNN	CNN	CNN-LSTM	LSTM-GRU
Accuracy	0.979	0.999	0.976	0.972	0.975	0.951
Precision	0.961	0.999	0.962	0.952	0.959	0.953
Recall	0.998	0.999	0.991	0.995	0.994	0.949
F1-score	0.979	0.999	0.976	0.973	0.976	0.951

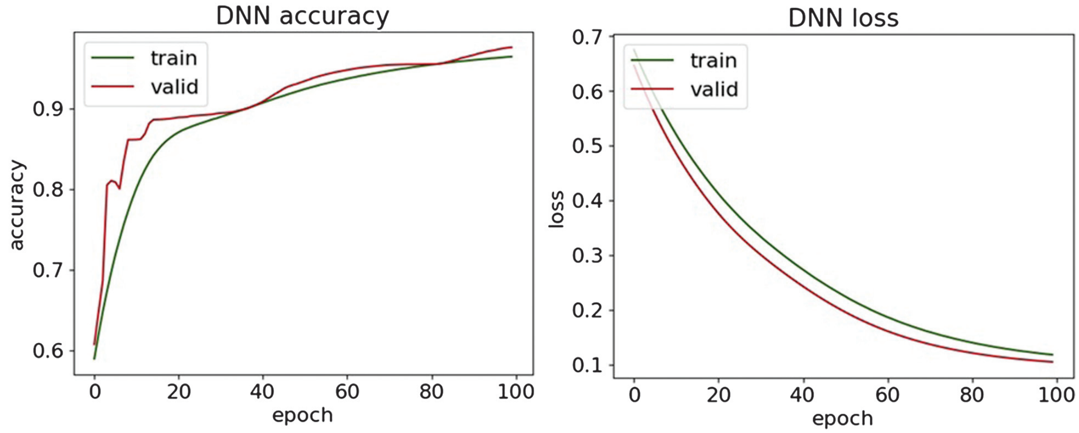


Fig. 13. The accuracy and loss plots of DNN.

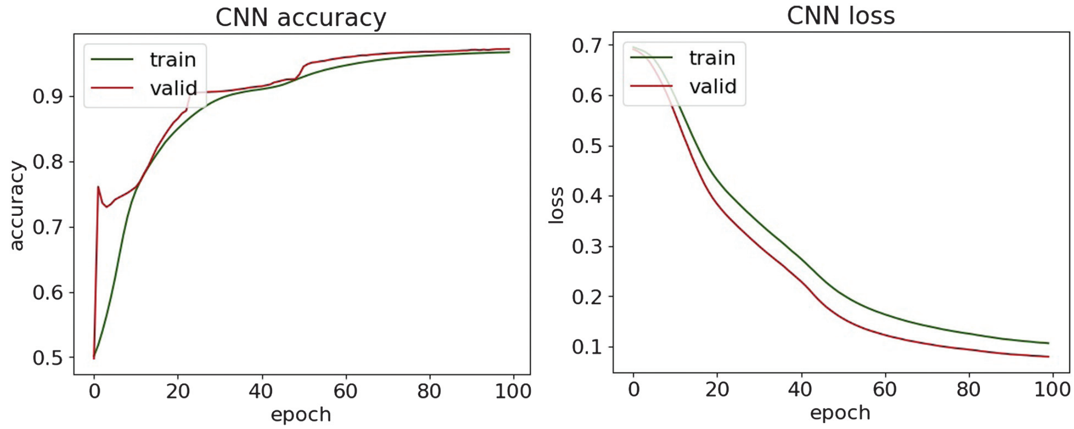


Fig. 14. The accuracy and loss plots of CNN.

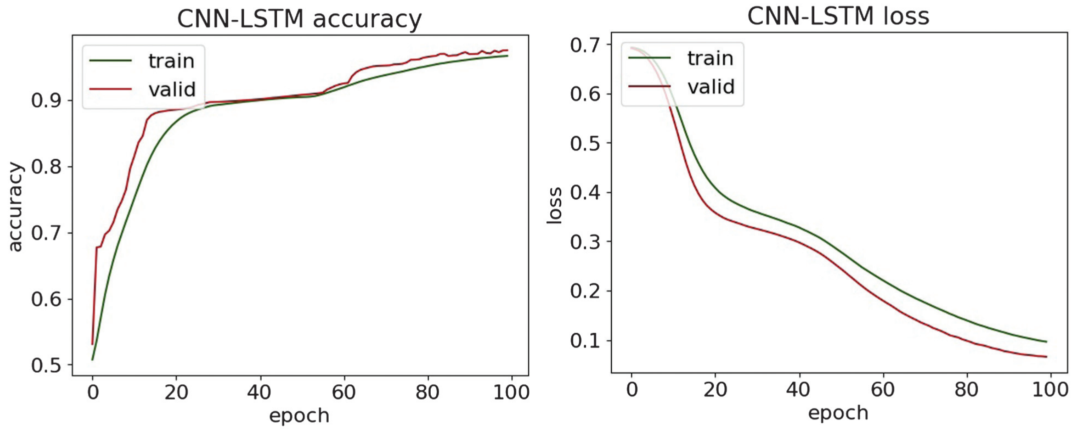


Fig. 15. The accuracy and loss plots of CNN-LSTM.

Additionally, the performance of the DL models was visually assessed through accuracy, loss, and confusion matrix plots. The accuracy plot illustrates the model's progression in correctly classifying samples during the training and validation phases, providing insight into how well the model learns over time. The loss plot tracked the model's error over time, helping to identify signs of convergence, underfitting, or overfitting by comparing training and validation loss curves. Additionally, confusion matrix plots were used to examine the distribution of true positives, true negatives, false positives, and false negatives for each model. These matrices offered a transparent and interpretable representation of the models' classification behavior, revealing whether they were biased toward certain classes or struggling with specific misclassifications. Together, these visualizations had a significant role in evaluating and comparing the effectiveness and reliability of each model in detecting cyber threats.

The classification results of DDoS attacks are shown in Table I. The accuracy, loss, and confusion matrices plots are presented in Fig. 13–17.

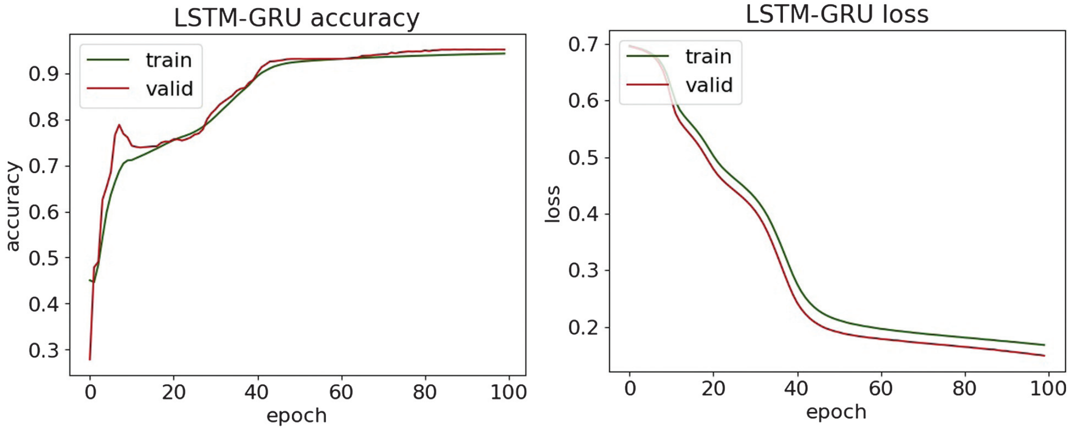


Fig. 16. The accuracy and loss plots of LSTM-GRU.

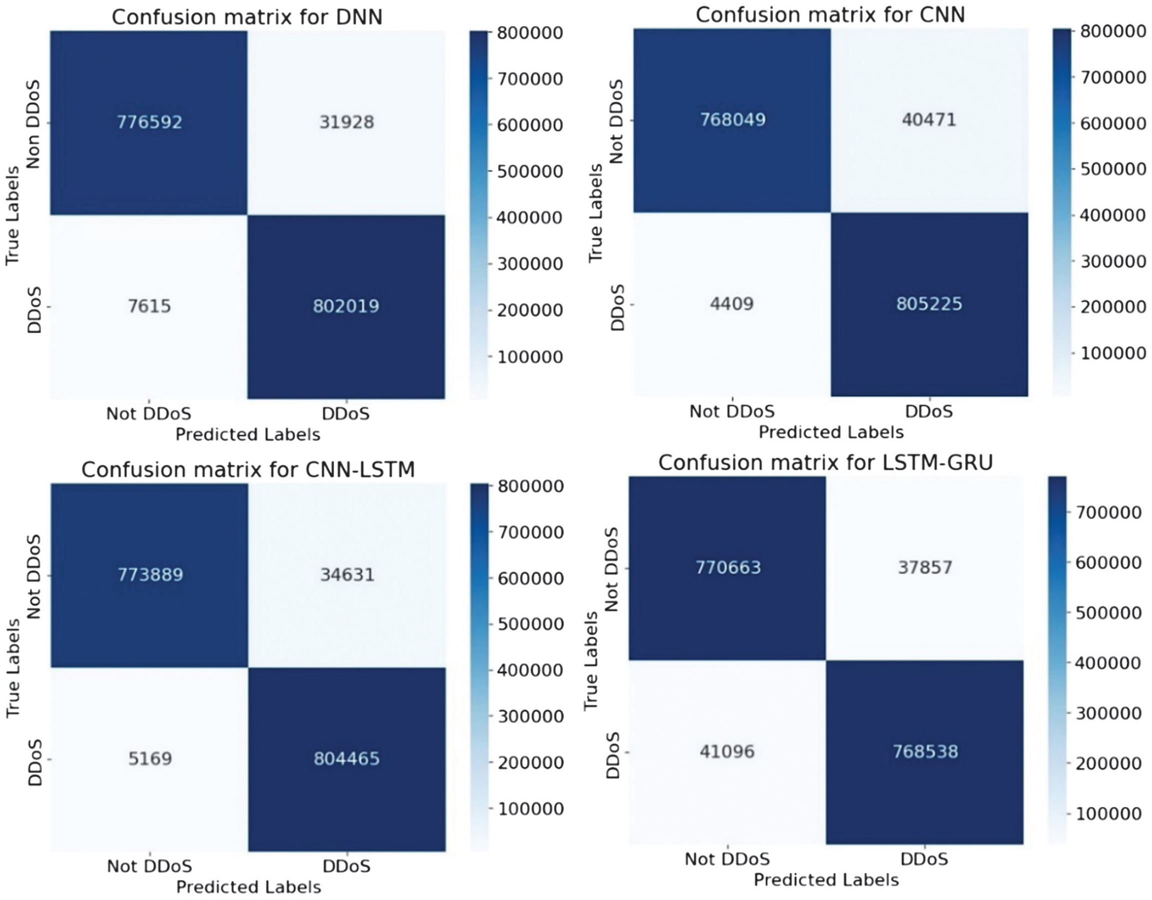


Fig. 17. The confusion matrices.

Table II. Classification results of MitM attacks

Metrics	SVM	Random forest	DNN	CNN	CNN-LSTM	LSTM-GRU
Accuracy	0.952	0.999	0.976	0.973	0.968	0.976
Precision	0.923	0.999	0.961	0.976	0.971	0.969
Recall	0.986	0.999	0.994	0.970	0.964	0.982
F1-score	0.953	0.999	0.977	0.973	0.967	0.976

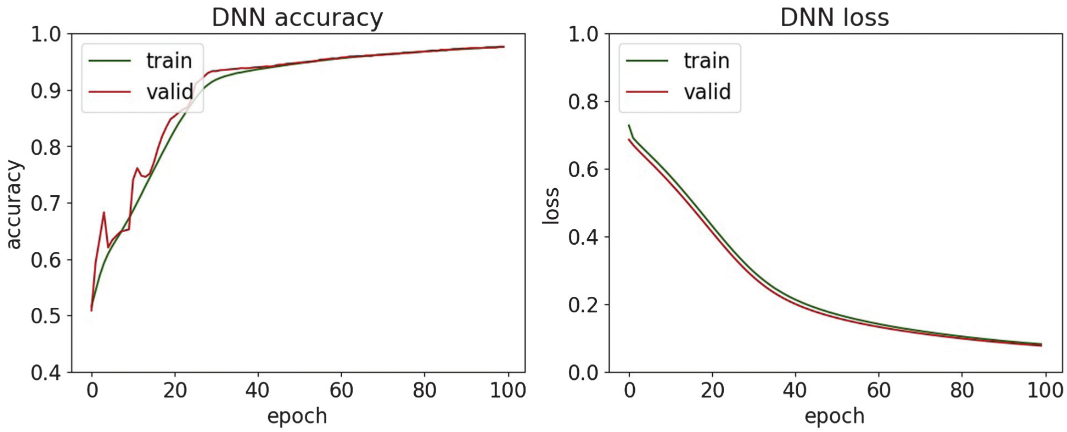


Fig. 18. The accuracy and loss plots of DNN.

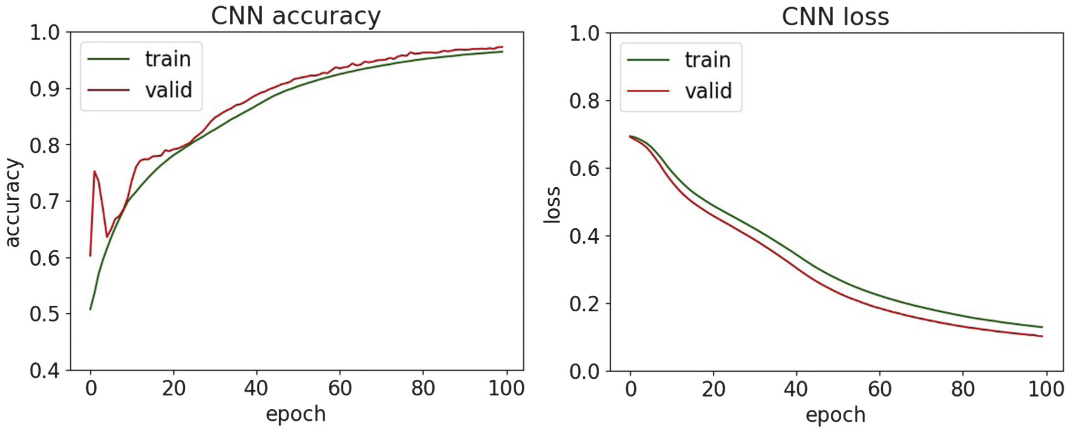


Fig. 19. The accuracy and loss plots of CNN.

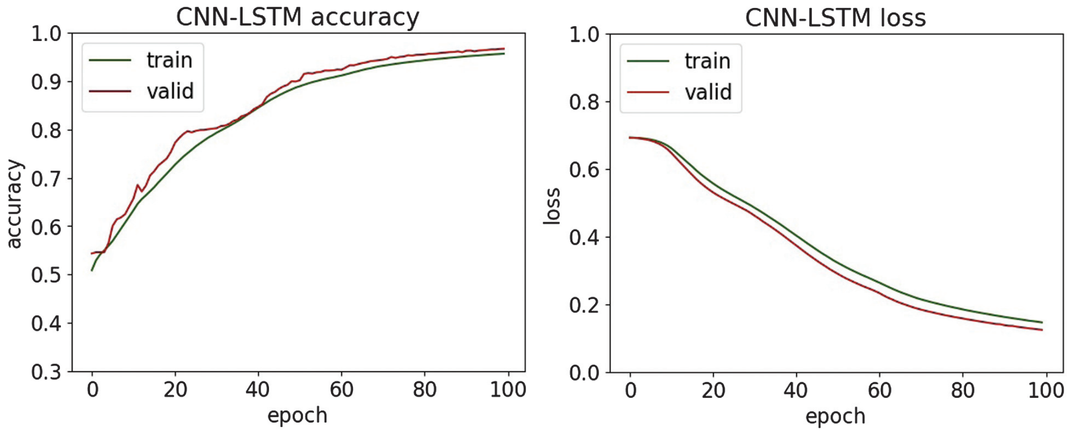


Fig. 20. The accuracy and loss plots of CNN-LSTM.

Along with the comparison of the received results themselves, they are evaluated with the results of the paper [36]. The obtained experimental results demonstrate superior detection performance, achieving accuracy values in the range of 95% and 97%, while the reported reference paper results have a maximum accuracy of 93.41% on a similar dataset. These findings highlight the effectiveness of the methods used in detecting the DDoS attack.

The classification results of MitM attacks are shown in Table II. The accuracy, loss, and confusion matrices plots are presented in Fig. 18–22.

The comparison of MitM classification results with the paper [37], which applies a multimodal Generative Adversarial Networks-enhanced detection framework, reports an accuracy score of approximately 83% on the MitM dataset. The presented research demonstrates substantially stronger detection. It delivers a targeted

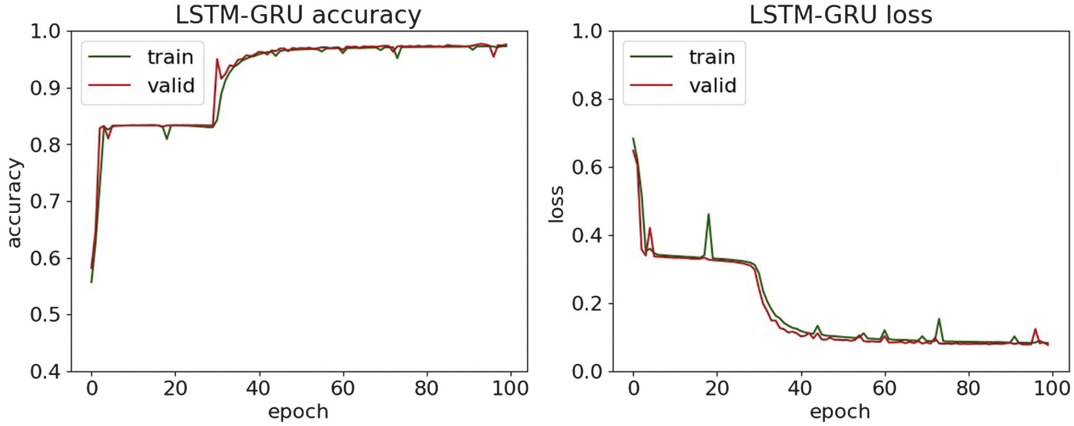


Fig. 21. The accuracy and loss plots of LSTM-GRU.

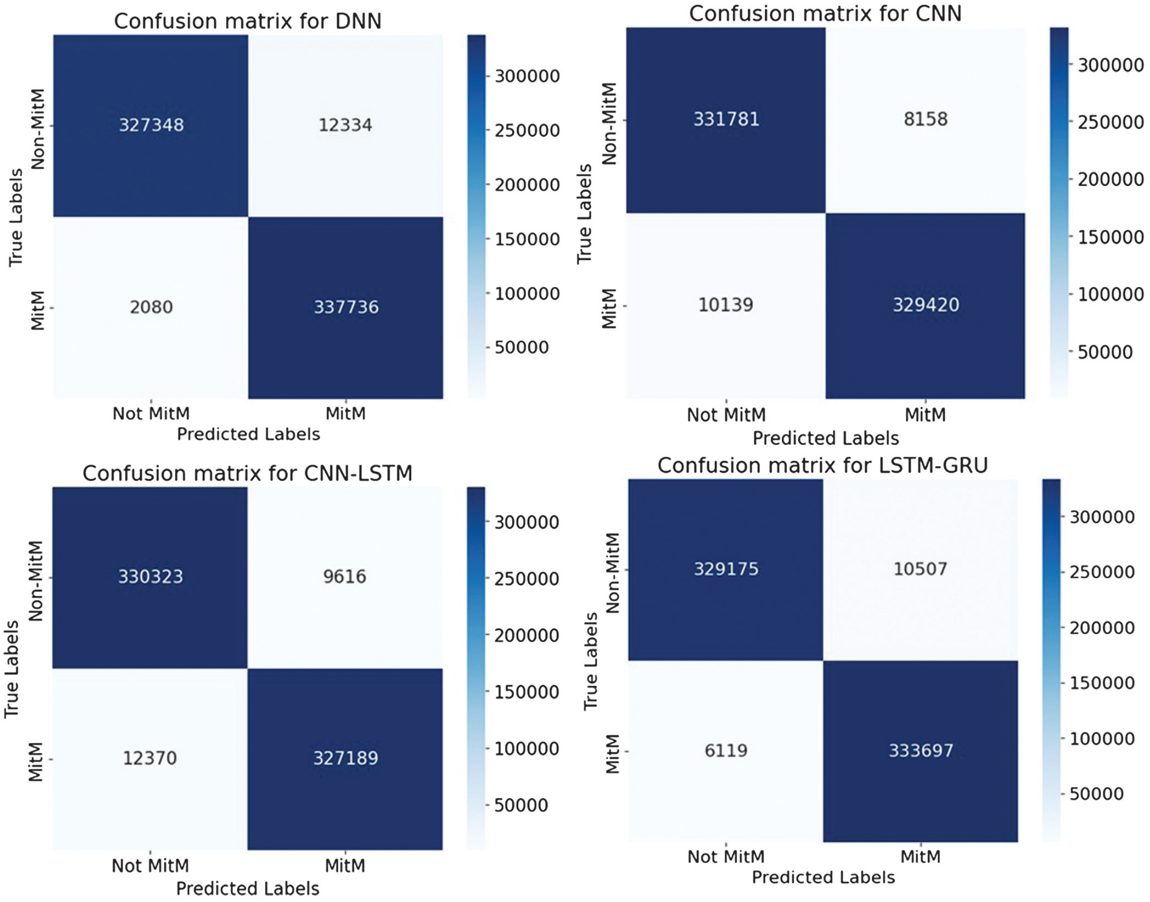


Fig. 22. The confusion matrices.

Table III. Classification results of malware

Metrics	SVM	Random forest	DNN	CNN	CNN-LSTM	LSTM-GRU
Accuracy	0.984	0.999	0.956	0.941	0.926	0.909
Precision	0.983	0.999	0.953	0.941	0.927	0.909
Recall	0.985	0.999	0.959	0.941	0.925	0.911
F1-score	0.984	0.999	0.956	0.941	0.926	0.910

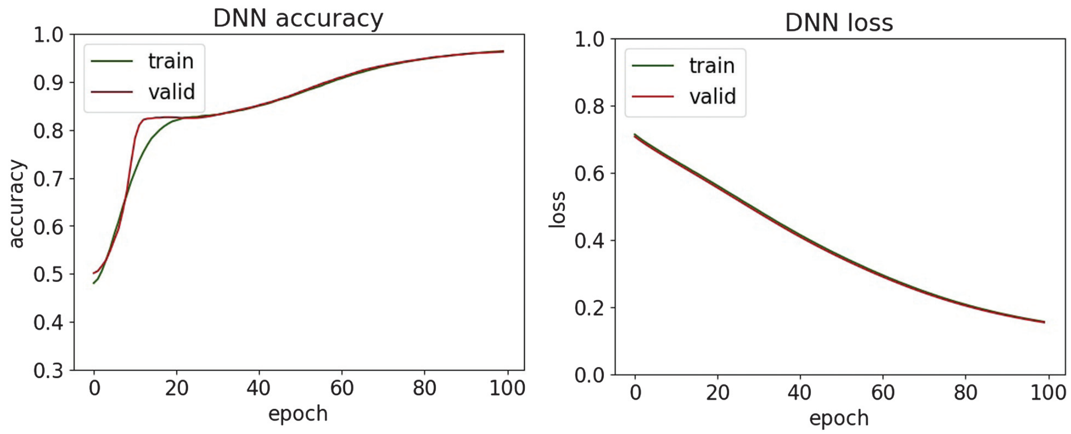


Fig. 23. The accuracy and loss plots of DNN.

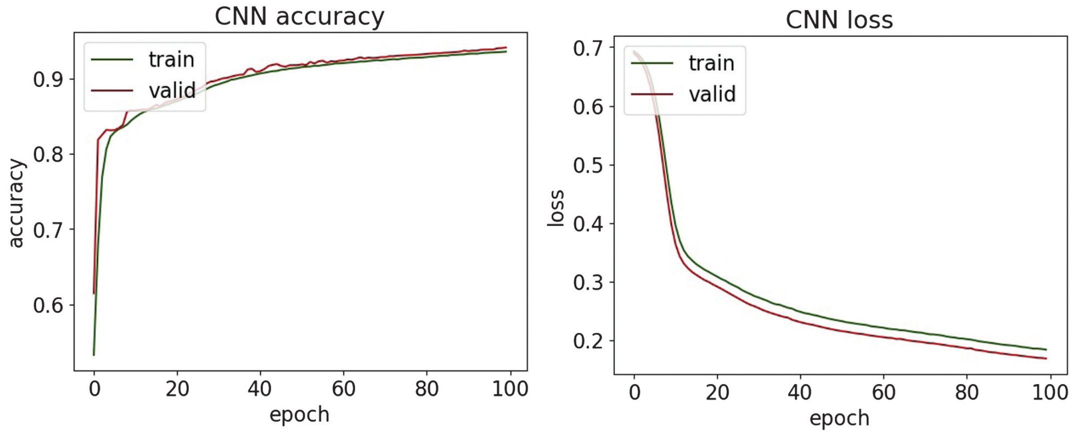


Fig. 24. The accuracy and loss plots of CNN.

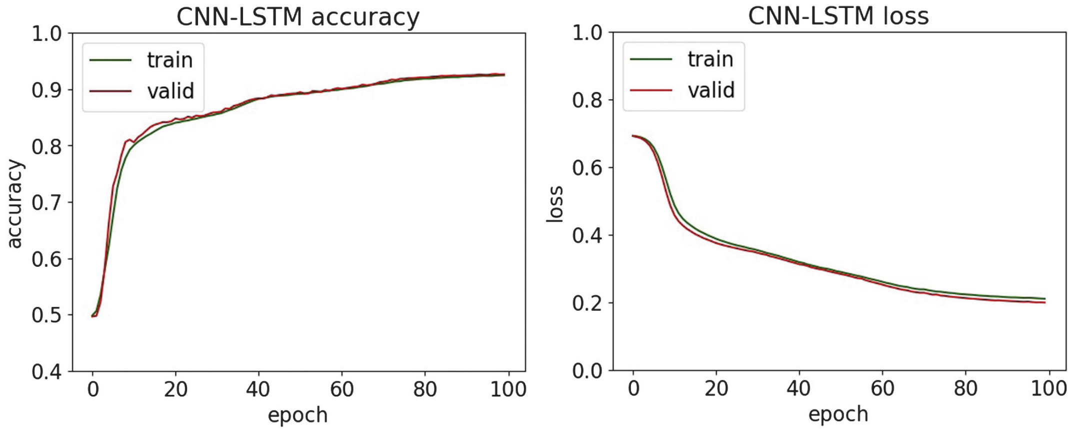


Fig. 25. The accuracy and loss plots of CNN-LSTM.

DL pipeline for MitM attacks, achieving an accuracy score of 97.6%. This combination of higher empirical performance and system-level implementation underscores the practical superiority and contribution of the presented study.

The classification results of malware attacks are shown in Table III. The accuracy, loss, and confusion matrices plots are presented in Fig. 23–27.

The malware classification results are compared with research [38], where the DL model achieves only a 69% accuracy score, indicating limited generalization capability for malware behavior patterns in this dataset. In contrast, the presented approach in the paper demonstrates substantially stronger performance, achieving above 94% accuracy on the same malware dataset category by efficient modern DL architectures and optimized training strategies. This comparison highlights that while previous work struggles to adapt deep models to this dataset effectively, this study advances the state of practice by delivering a significantly more robust malware detection pipeline.

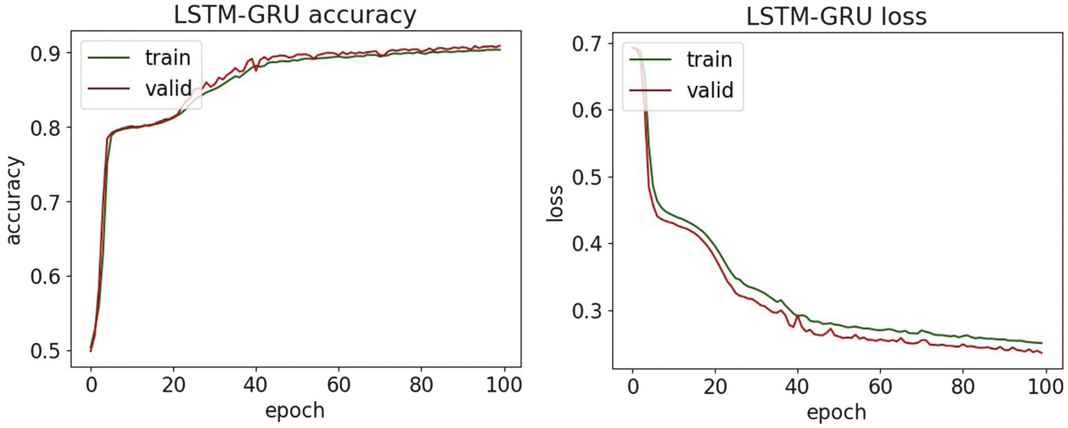


Fig. 26. The accuracy and loss plots of LSTM-GRU.

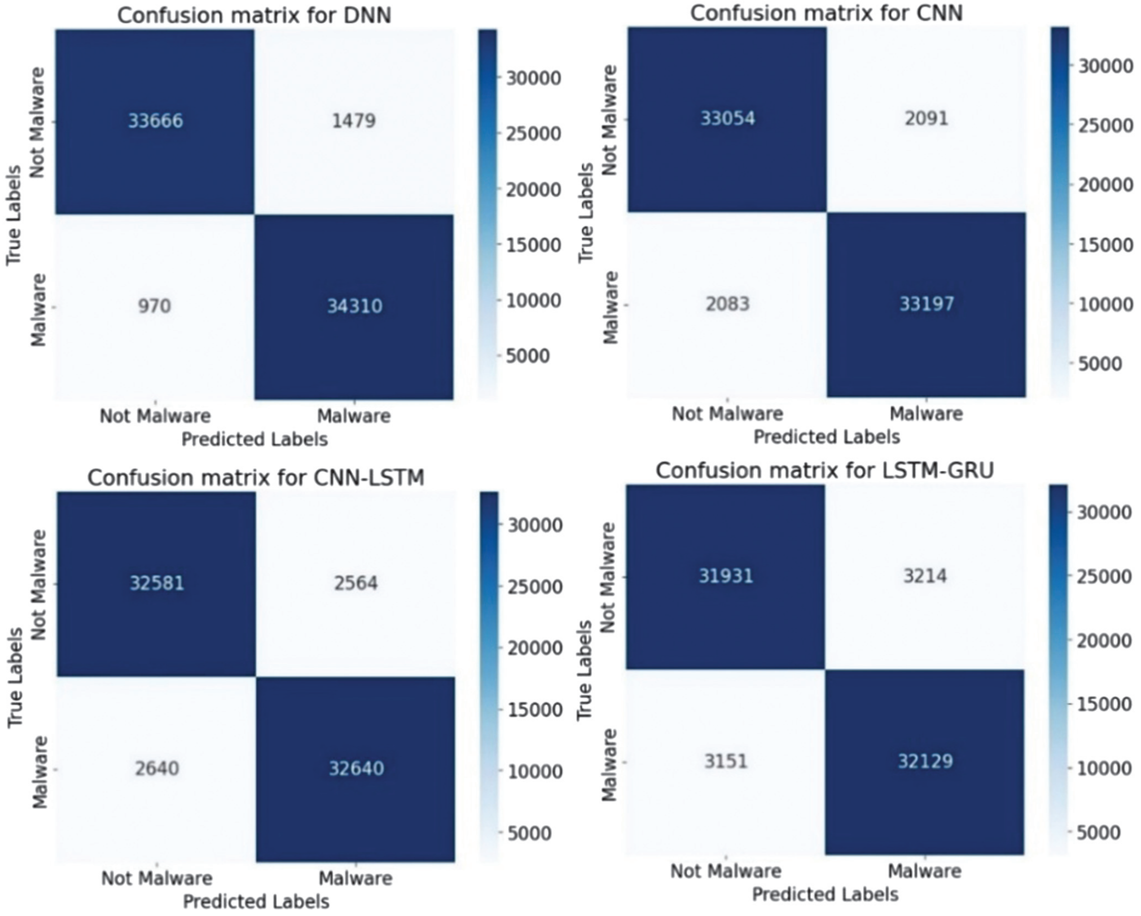


Fig. 27. The confusion matrices.

Table IV. Classification results of phishing

Metrics	SVM	Random forest	DNN	CNN	CNN-LSTM	LSTM-GRU
Accuracy	0.812	0.846	0.817	0.622	0.628	0.735
Precision	0.873	0.913	0.881	0.776	0.737	0.774
Recall	0.732	0.766	0.733	0.341	0.395	0.662
F1-score	0.797	0.833	0.801	0.474	0.515	0.714

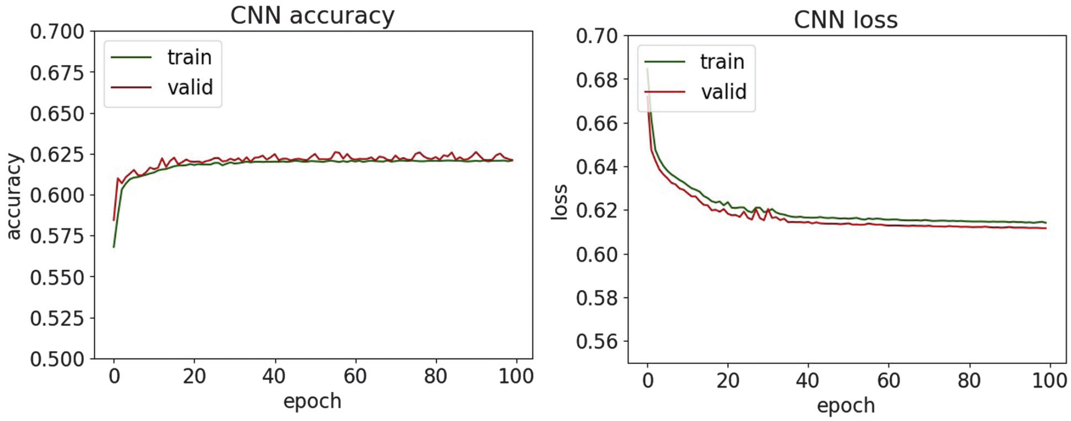


Fig. 28. The accuracy and loss plots of DNN.

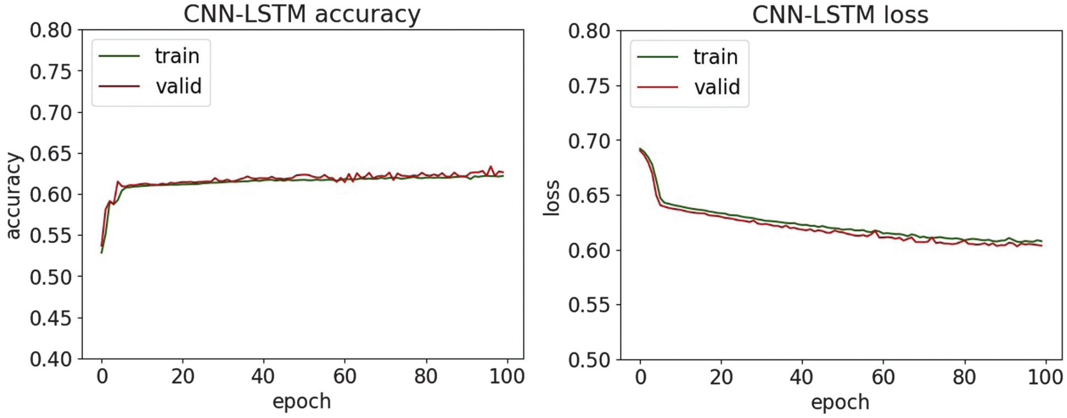


Fig. 29. The accuracy and loss plots of CNN.

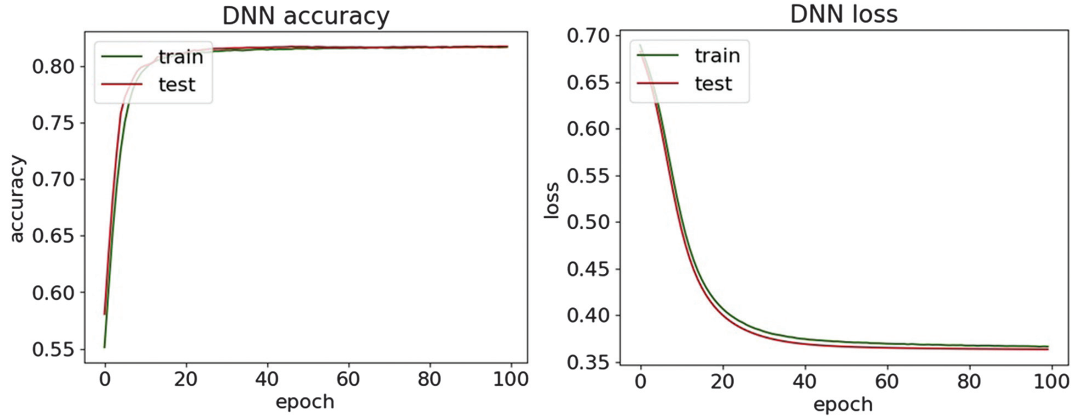


Fig. 30. The accuracy and loss plots of CNN-LSTM.

The classification results of phishing are shown in Table IV. The accuracy, loss, and confusion matrices plots are presented in Fig. 28–32.

In [39], several classical ML classifiers are evaluated on a large phishing dataset and report the highest accuracy of **76%**. On the contrary, the presented research leverages DL architectures, such as CNN-LSTM and LSTM-GRU, and applies enhanced feature extraction techniques tailored for URL patterns, achieving **over 81% accuracy** on the same dataset. Therefore, the presented method pushes performance significantly higher, establishing a new benchmark for phishing detection.

The classification results of SQL injection are shown in Table V. The accuracy, loss, and confusion matrices plots are presented in Fig. 33–37.

In the study [40], there is a CNN for feature extraction and an RF classifier for decision-making to detect SQL injection attacks and report an accuracy of **75.6%** under their experimental setup. By contrast, the presented research reaches higher accuracy on the

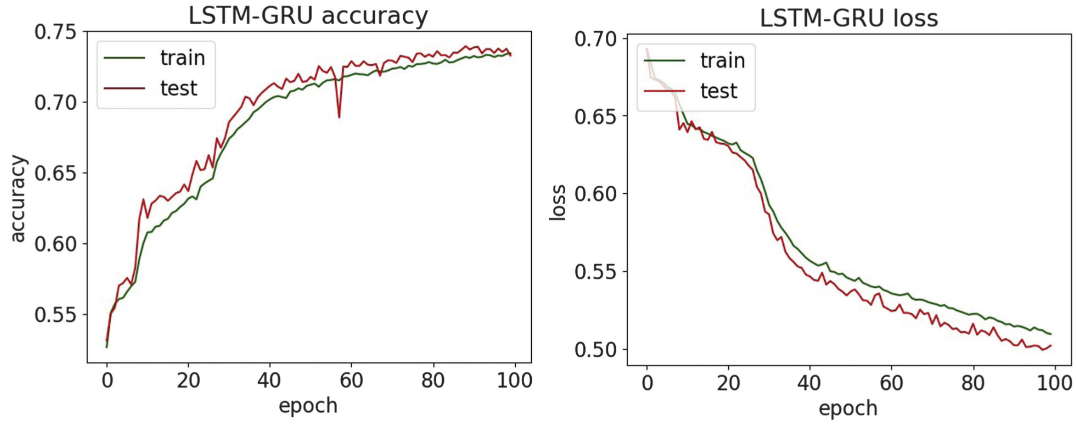


Fig. 31. The accuracy and loss plots of LSTM-GRU.

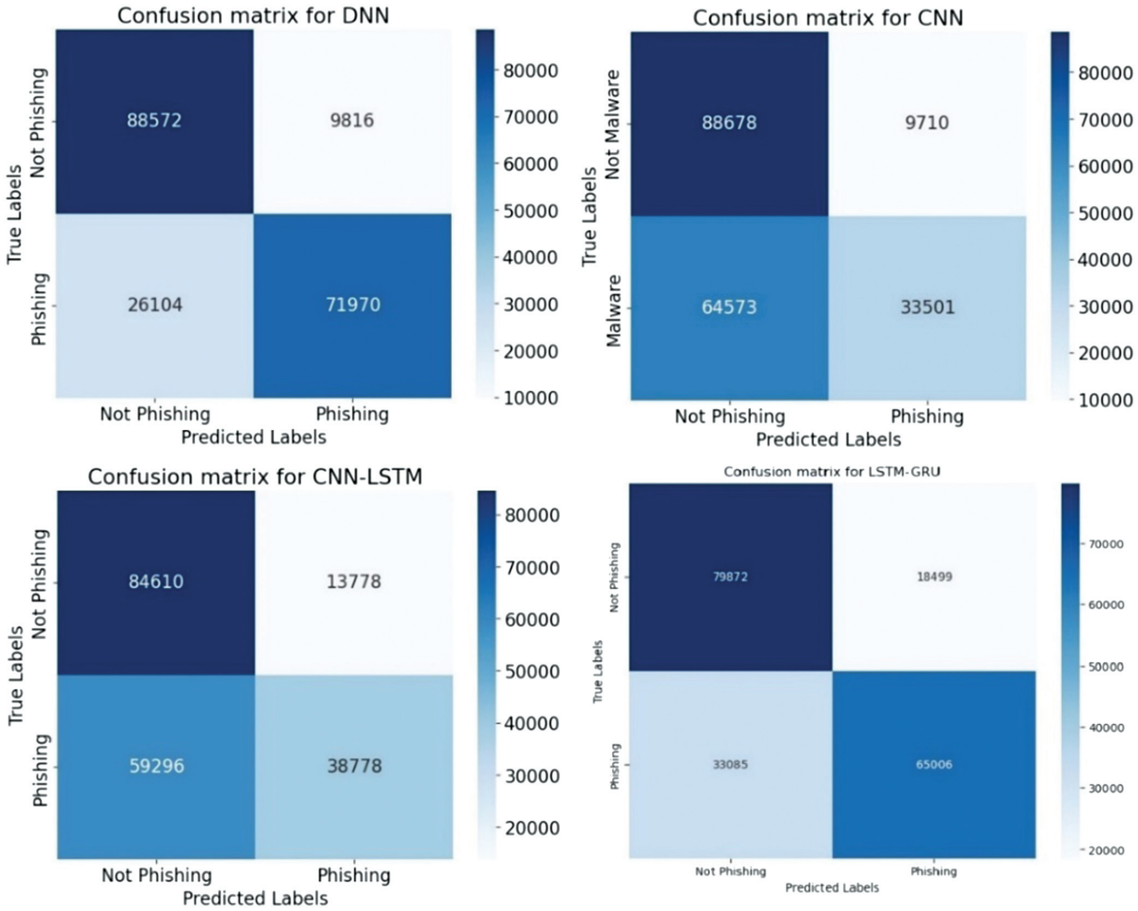


Fig. 32. The confusion matrices.

Table V. Classification results of SQL injection

Metrics	SVM	Random forest	DNN	CNN	CNN-LSTM	LSTM-GRU
Accuracy	0.908	0.951	0.911	0.785	0.787	0.784
Precision	0.954	0.985	0.942	0.766	0.761	0.823
Recall	0.856	0.917	0.877	0.823	0.840	0.724
F1-score	0.903	0.950	0.908	0.793	0.799	0.770

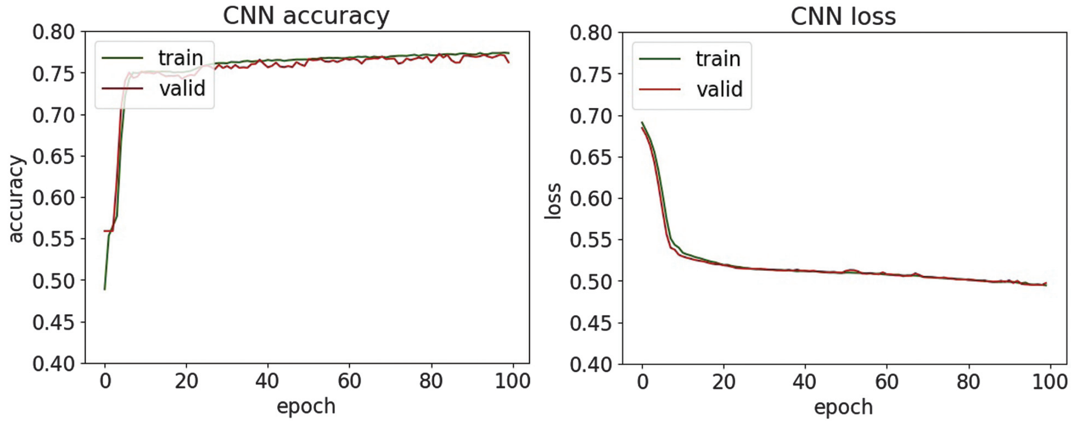


Fig. 33. The accuracy and loss plots of DNN.

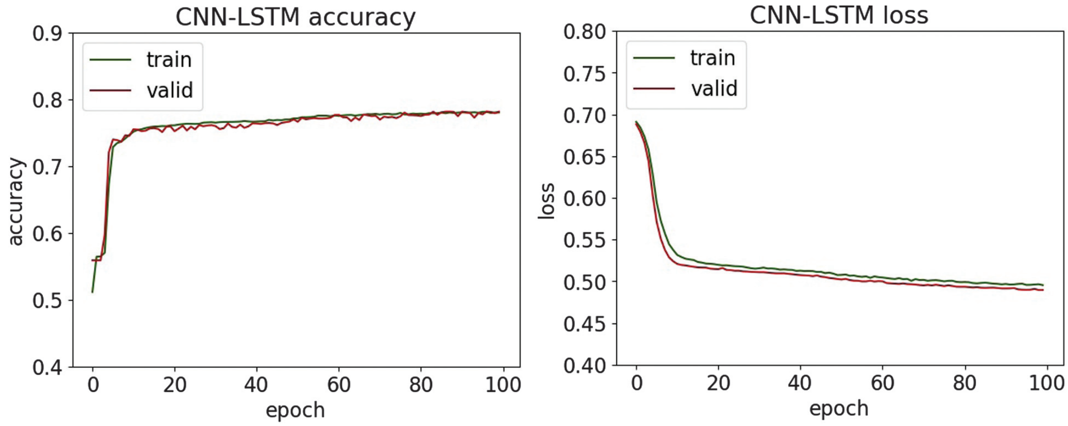


Fig. 34. The accuracy and loss plots of CNN.

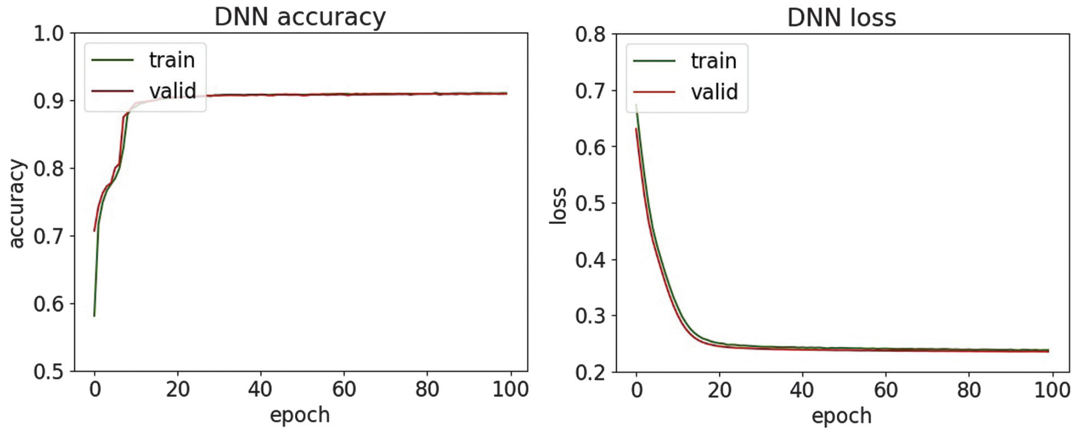


Fig. 35. The accuracy and loss plots of CNN-LSTM.

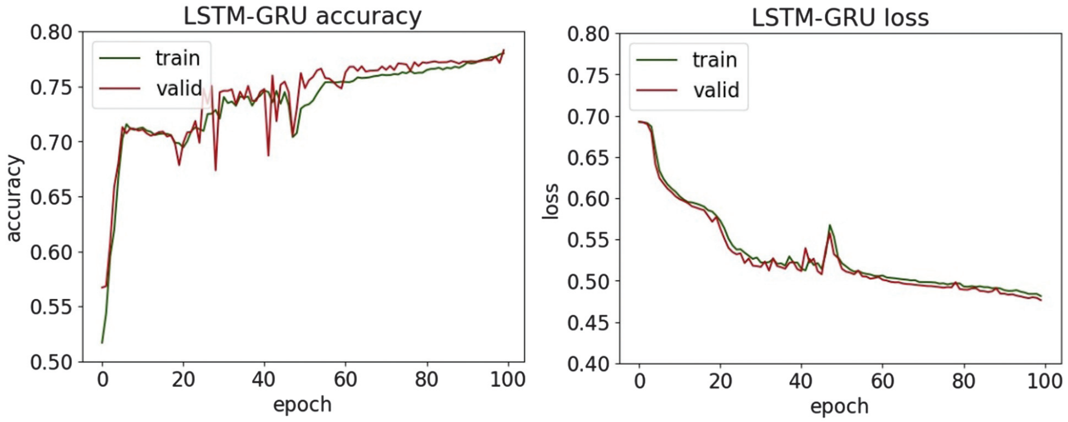


Fig. 36. The accuracy and loss plots of LSTM-GRU.

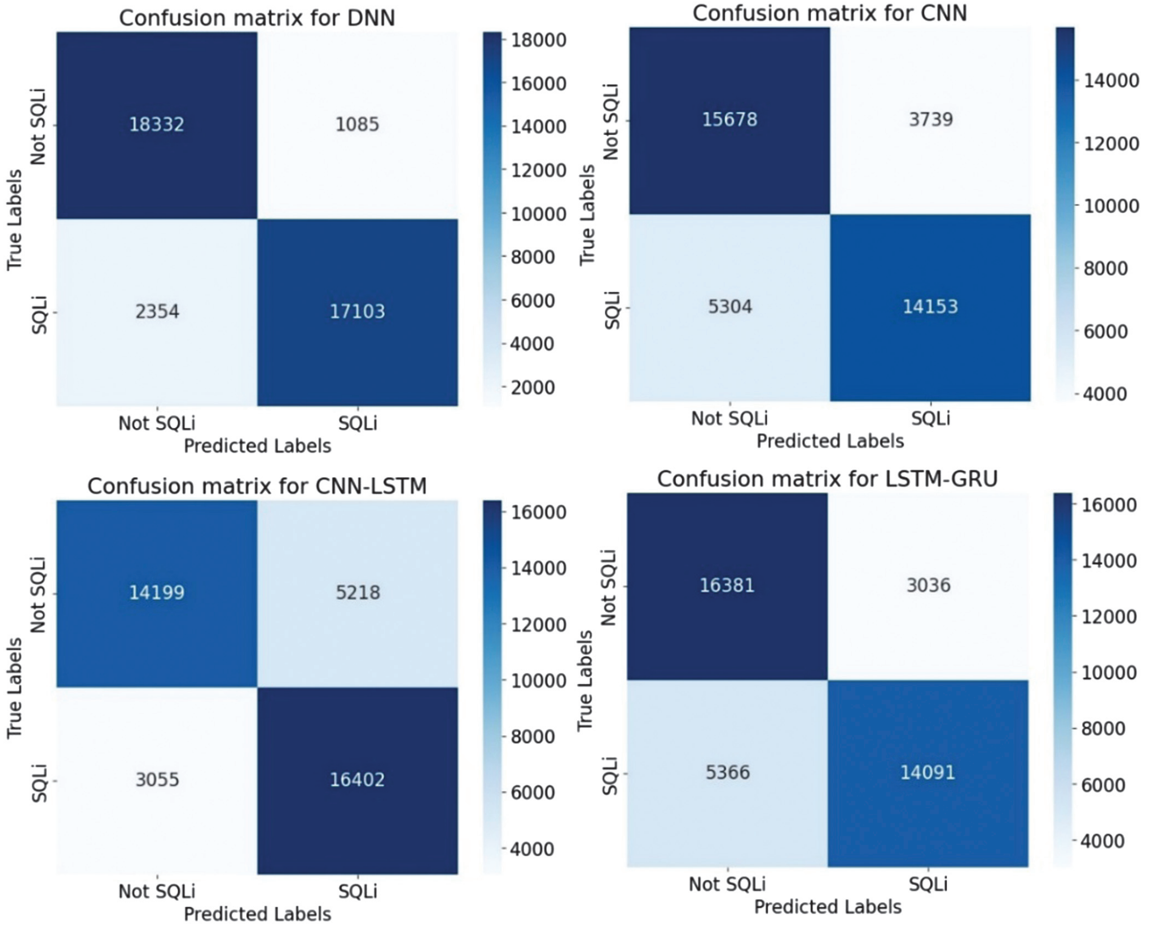


Fig. 37. The confusion matrices.

same or comparable SQL injection dataset and demonstrates classification accuracies of up to **95%**. This shows that our model surpasses the previously reported benchmark, thereby offering improved detection performance and broader applicability in security contexts.

An experimental evaluation of the proposed DL models highlights both the strengths and limitations of each architecture in detecting various types of cyber threats. Across all datasets, DNN and CNN-LSTM demonstrate consistently high accuracy, particularly in classifying structured network threats, such as DDoS and MITM attacks. It suggests that the fully connected nature of DNN and the temporal–spatial processing capabilities of CNN-LSTM are well suited to modeling regular patterns in network traffic. For DDoS detection, all four models achieved accuracy above 95%, with DNN and CNN-LSTM achieving the highest F1-scores of 97.6%. It shows an ability to distinguish between legitimate and malicious traffic, which is essential for real-time defense mechanisms. Similar trends are observed in MitM classification, where the precision and recall metrics further validate the robustness of the models. Interestingly, LSTM-GRU showed competitive performance, reflecting the usefulness of combining memory-based architectures for consistent attack detection. For malware classification, performance dropped slightly, although DNN and CNN were still able to maintain precision and recall above 94%. The drop in overall precision suggests that malware features may be more diverse and more challenging to generalize, requiring deeper or more specialized networks.

On the other hand, phishing and SQL injection attacks—both text-based and less structured—presented a bigger challenge. Here, precision dropped below 82% for most models, with CNN and CNN-LSTM having the most difficulty. This drop in performance means that models trained on sequential and spatial data may not generalize well to raw text unless enhanced with natural language processing techniques. Despite these issues, the high precision values on the phishing and SQL injection tasks indicate that when the model identifies a threat, it is usually correct. However, it may still miss many real-world threats, resulting in lower recall. This trade-off suggests potential for improvement through data augmentation or hybrid systems that combine DL with rule-based methods. Visual analysis using accuracy and loss plots, as well as confusion matrices, provided further insight into the model behavior. In particular, DNN demonstrated robust convergence across all datasets, whereas recurrent models, such as LSTM-GRU, showed greater variance in loss curves, possibly due to their sensitivity to sequence length and temporal noise. These results highlight the need for fine-tuning hyperparameters and more rigorous application of regularization methods for recurrent models. Overall, the experimental results support the applicability of DL for intrusion detection in healthcare cybersecurity. However, further optimization, especially in text-heavy threat categories, remains necessary to build a comprehensive and robust security solution.

V. CONCLUSION

The research focused on the classification of five major cybersecurity threats: DDoS, MitM, malware, phishing, and SQLi, using DL models, including DNN, CNN, CNN-LSTM, and LSTM-GRU. The study addressed the increasingly complex cybersecurity landscape within healthcare institutions, where protecting sensitive patient data was critical. An important element of this research was the implementation of a secure hardware–software architecture that leveraged WireGuard, a lightweight and modern VPN protocol. WireGuard was employed to establish encrypted tunnels (e.g., wg0 and wg1) across network nodes, ensuring secure data communication and robust routing of cryptographic keys. Its configuration utilized conventional network tools such as ifconfig, ip, and route, facilitating a seamless and secure communication framework. WireGuard's approach to associating each peer with public keys and internal IP addresses significantly enhanced authentication and data exchange security. The threat classification part covered a comprehensive methodology, comprising dataset collection and preprocessing, feature extraction, model training, and performance evaluation. The preprocessing involved specific techniques tailored to the nature of each dataset, including min–max normalization for structured network data and TF-IDF for text-based threats. Chi-square feature selection identified the most significant features for enhancing classification accuracy. For structured threats like DDoS and MitM attacks, all models showed robust performance. Specifically, DNN and CNN-LSTM consistently achieved the highest accuracy and F1-scores, approximately 97.6%, highlighting their effectiveness in distinguishing malicious from legitimate network traffic. LSTM-GRU also showed competitive performance in MitM attacks, reflecting the strength of memory-based architectures in detecting temporal attack patterns. In malware classification tasks, DNN maintained the best performance, achieving 95.6% accuracy, followed closely by CNN. However, overall accuracy and precision slightly declined, indicating the complex and heterogeneous nature of malware features that necessitate deeper or more specialized models. Conversely, detecting less structured, text-based threats such as phishing and SQL injection presented greater challenges. Accuracy and precision notably decreased, with precision dropping below 82% in most cases, especially for CNN and CNN-LSTM models. This reduction suggested limitations in handling textual and sequential data without advanced natural language processing techniques. Despite this, models maintained high precision when identifying actual threats, albeit with lower recall, indicating missed threats. Visual analyses, including accuracy and loss plots alongside confusion matrices, provided deeper insights into model behaviors. DNN consistently demonstrated stable convergence across all tasks, indicating strong generalization capabilities. Recurrent architectures like LSTM-GRU showed higher variability, emphasizing their sensitivity to temporal dynamics and sequence lengths and thus highlighting the importance of careful hyperparameter tuning and regularization. Overall, the results confirmed the viability of DL models for cybersecurity in healthcare, particularly for structured network attacks. However, further optimization and incorporation of sophisticated natural language processing or hybrid approaches are necessary to effectively handle more complex, text-based threats and build a comprehensive security solution.

ACKNOWLEDGMENT

This research was carried out within the framework of the project AP19675957, "The research and development of the system for ensuring the protection of medical data using blockchain technology and artificial intelligence methods," which is being implemented at the Institute of Information and Computer Technologies.

CONFLICT OF INTEREST STATEMENT

The author(s) declared no potential conflicts of interest in the publication of this article.

REFERENCES

[1] I. Hamid and M. M. H. Rahman, "AI, machine learning and deep learning in cyber risk management," *Discover Sustainability*, vol. 6, p. 389, 2025. https://doi.org/10.1007/s43621-025-01012-3

[2] R. Haripriya *et al.*, "A privacy-enhanced framework for collaborative big data analysis in healthcare using adaptive federated learning aggregation," *J. Big Data.*, vol. 12, p. 113, 2025. https://doi.org/10.1186/s40537-025-01169-8

[3] L. Wang *et al.*, "SIFT: Enhance the performance of vulnerability detection by incorporating structural knowledge and multi-task learning," *Autom. Softw. Eng.*, vol. 32, p. 38, 2025. https://doi.org/10.1007/s10515-025-00507-7

[4] P. Li and L. Zhang, "Application of big data technology in enterprise information security management," *Sci. Rep.*, vol. 15, p. 1022, 2025. https://doi.org/10.1038/s41598-025-85403-6

[5] M. Husain Bathushaw and S. Nagasundaram, "The role of blockchain and AI in fortifying cybersecurity for healthcare systems," *Int. J. Comput. Exp. Sci. Eng.*, vol. 10, no. 4, pp. 1120–1129, 2024. DOI: https://doi.org/10.22399/ijcesen.596

[6] N. Sivanesan *et al.*, "Comparison of mitigating DDoS attacks in software defined networking and IoT platforms," *Cyber. Secur. Appl.*, vol. 3, p. 100080, 2025. https://doi.org/10.1016/j.csa.2024.100080

[7] M. A. Ali and S. A. H. Al-Sharafi, "Intrusion detection in IoT networks using machine learning and deep learning approaches for MitM attack mitigation," *Discover Internet Things*, vol. 5, p. 48, 2025. https://doi.org/10.1007/s43926-025-00104-w

[8] E. Baghirov, "A comprehensive investigation into robust malware detection with explainable AI," *Cyber. Secur. Appl.*, vol. 3, p. 100072, 2025. https://doi.org/10.1016/j.csa.2024.100072

[9] J. Zhou *et al.*, "An integrated CSPPC and BiLSTM framework for malicious URL detection," *Sci. Rep.*, vol. 15, p. 6659, 2025. https://doi.org/10.1038/s41598-025-91148-z

[10] Y. Chen, G. Liang, and Q. Wang, "Research on SQL injection detection technology based on content matching and deep learning," *Comput. Mater. Contin.*, vol. 84, no. 1, pp. 1145–1167, 2025. https://doi.org/10.32604/cmc.2025.063319

[11] V. Kanimozhi and T. P. Jacob, "The top ten artificial intelligence-deep neural networks for IoT intrusion detection system," *Wirel. Pers. Commun.*, vol. 129, pp. 1451–1470, 2023. https://doi.org/10.1007/s11277-023-10198-6

[12] Tolani K. C. *et al.*, "Cybersecurity challenges in AI-driven energy systems: Current and future prospects concerning ethical and legal provisions," in Leveraging AI for Innovative Sustainable Energy: Solar, Wind and Green Hydrogen, H. Hammouch and L. Razzak Janjua (Eds.), Pennsylvania, USA: IGI Global Scientific Publishing, pp. 109–122, 2025. https://doi.org/10.4018/979-8-3373-0045-0.ch008

[13] E. B. Blancaflor *et al.*, "Advanced phishing techniques: Analyzing adversary-in-the-middle and browser-in-the-browser attacks in modern cybersecurity," *Cybern. Inf. Technol.*, vol. 25, no. 1, pp. 55–77, 2025. https://doi.org/10.2478/cait-2025-0004

[14] B. Brindavathi, A. Karrothu, and C. Anilkumar, "An analysis of AI-based SQL injection (SQLi) attack detection," *Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, Trichy, India, pp. 31–35, 2023. https://doi.org/10.1109/ICAISS58487.2023.10250505

[15] K. Qi, "Advancing hospital healthcare: Achieving IoT-based secure health monitoring through multilayer machine learning," *J. Big Data*, vol. 12, p. 1, 2025. https://doi.org/10.1186/s40537-024-01038-w

[16] M. Kumari, M. Gaikwad, and S. A. Chavan, "A secure IoT-edge architecture with data-driven AI techniques for early detection of cyber threats in healthcare," *Discover Internet Things*, vol. 5, p. 54, 2025. https://doi.org/10.1007/s43926-025-00147-z

[17] A. Aly *et al.*, "Real-time multi-class threat detection and adaptive deception in Kubernetes environments," *Sci. Rep.*, vol. 15, p. 8924, 2025. https://doi.org/10.1038/s41598-025-91606-8

[18] M. Nadeem and C. Hongsong, "Advancing social network security with magteon-turing L3TM: A multi-layered defense system against cyber threats," *Comput. Netw.*, vol. 267, p. 111375, 2025. https://doi.org/10.1016/j.comnet.2025.111375

[19] J. Cisneros-Gonzalez *et al.*, "JorGPT: Instructor-aided grading of programming assignments with large language models (LLMs)," *Future Internet*, vol. 17, p. 265, 2025. https://doi.org/10.3390/fi17060265

[20] E. Brito *et al.*, "Decentralized proof-of-location systems for trust, scalability, and privacy in digital societies," *Sci. Rep.*, vol. 15, p. 19808, 2025. https://doi.org/10.1038/s41598-025-04566-4

[21] A. Bensaoud and J. Kalita, "CleanSheet: Advancing backdoor attack techniques for deep neural networks with stealthy trigger embedding," *Syst. Soft Comput.*, vol. 7, p. 200335, 2025. https://doi.org/10.1016/j.sasc.2025.200335

[22] A. K. B. Arnob *et al.*, "An enhanced LSTM approach for detecting IoT-based DDoS attacks using honeypot data," *Int. J. Comput. Intell. Syst.*, vol. 18, p. 19, 2025. https://doi.org/10.1007/s44196-025-00741-7

[23] M. A. Ali and S. A. H. Al-Sharafi, "Intrusion detection in IoT networks using machine learning and deep learning approaches for MitM attack mitigation," *Discover Internet Things*, vol. 5, p. 48, 2025. https://doi.org/10.1007/s43926-025-00104-w

[24] N. Karmous *et al.*, "Deep learning approaches for protecting IoT devices in smart homes from MitM attacks," *Front. Comput. Sci.*, vol. 6, p. 1477501, 2024. https://doi.org/10.3389/fcomp.2024.1477501

[25] V. Kandasamy and A. A. Roseline, "Harnessing advanced hybrid deep learning model for real-time detection and prevention of man-in-the-middle cyber attacks," *Sci. Rep.*, vol. 15, p. 1697, 2025. https://doi.org/10.1038/s41598-025-85547-5

[26] H. Bakır, "A new method for tuning the CNN pre-trained models as a feature extractor for malware detection," *Pattern Anal. Appl.*, vol. 28, p. 26, 2025. https://doi.org/10.1007/s10044-024-01381-x

[27] S. Mousavi and M. Bahaghighat, "Phishing website detection: An in-depth investigation of feature selection and deep learning," *Expert Syst.*, vol. 42, no. 3, pp. 1–29, 2025. https://doi.org/10.1111/exsy.13824

[28] A. Alhuzali *et al.*, "In-depth analysis of phishing email detection: Evaluating the performance of machine learning and deep learning models across multiple datasets," *Appl. Sci.*, vol. 15, no. 6, 3396, 2025. https://doi.org/10.3390/app15063396

[29] S. F. Kholood, B. Sherif, and A. Rezk, "An effective SQL injection detection model using LSTM for imbalanced datasets," "*Comput. Secur.*, vol. 153, p. 104391, 2025. https://doi.org/10.1016/j.cose.2025.104391

[30] D. Muduli *et al.*, "SIDNet: A SQL injection detection network for enhancing cybersecurity," IEEE Access, vol. 12, pp. 176511–176526, 2024. https://doi.org/10.1109/ACCESS.2024.3502293

[31] O. Ussatova *et al.*, *Comprehensive DDoS Attack Classification Using Machine Learning Algorithms, Computer, Materials & Continua. Tech Science Press.* vol. 73, no. 1, pp. 577–594, 2022. https://doi.org/10.32604/cmc.2022.026552

[32] O. Ussatova *et al.*, "The development of a model for the threat detection system with the use of machine learning and neural network methods," *Int. J. Innov. Res. Sci. Stud. (IJIRSS)*, vol. 7, no. 3, pp. 863–877, 2024. https://doi.org/10.53894/ijirss.v7i3.2957

[33] A. Zhumabekova *et al.*, "The Application of Machine and Deep Learning Models for Detecting Malware Threats in Healthcare Organizations," *5th International Conference on Advanced Research in Computing (ICARC)*, February 19-20, Belihuloya, Sri Lanka, 227, pp. 1–6, 2025. https://doi.org/10.1109/ICARC64760.2025.10962965

[34] W. Ke *et al.*, "Improving the transferability of adversarial examples through neighborhood attribution," *Knowl.-Based Syst.*, vol. 296, p. 111909, 2024. https://doi.org/10.1016/j.knosys.2024.111909

[35] D. Zheng *et al.*, "Enhancing the transferability of adversarial attacks via multi-feature attention," *Trans. Inf. Forensics Secur.*, vol. 20, pp. 1462–1474, 2025. https://doi.org/10.1109/TIFS.2025.3526067

[36] A. Sharmin and Y. N. Abdullah, “Towards DDoS attack detection using deep learning approach,” *Comput. Secur.*, vol. 129, pp. 1–15, 2023. https://doi.org/10.1016/j.cose.2023.103251

[37] N. Peppes *et al.*, “A multimodal framework for advanced cybersecurity threat detection using GAN-driven data synthesis,” *Appl. Sci.*, vol. 15, p. 8730, 2025. https://doi.org/10.3390/app15158730

[38] F. Sibarani and P. Chan, “A comparative study of machine learning and deep learning algorithms for malware detection,” *J. Comput. Sci. Technol. Stud.*, vol. 7, no. 9, pp. 636–651, 2025. https://doi.org/10.32996/jcsts.2025.4.1.75

[39] K. Owa and O. Adewole, “Benchmarking machine learning techniques for phishing detection and secure URL classification,” *Int. J. Comput. Sci. Mob. Comput.*, vol. 14, no. 1, pp. 20–37, 2025. https://doi.org/10.47760/ijcsmc.2025.v14i01.003

[40] S. R. Menaka *et al.*, “An efficient SQL injection detection with a hybrid CNN & random forest approach,” *J. Inf. Syst. Eng. Manage.*, vol. 10, no. 18, pp. 1–10, 2025. https://doi.org/10.52783/jisem.v10i18s.2979