iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VII Month of publication: July 2025

DOI: https://doi.org/10.22214/ijraset.2025.73355

Classification of Threats Using Deep Learning Algorithms

Koda Prabhuteja[1], Challa Narasimham[2]
[1]MCA student, [2]IOCL – Chair Professor, Department of Information Technology & Computer Applications, Andhra University College of Engineering, Visakhapatnam, AP

Abstract: Cybersecurity threats are becoming more complex and frequent, making traditional detection methods insufficient. To address this, the project introduces a deep learning-based system for automatically detecting and classifying network threats. It uses structured traffic data from the CyberFedDefender dataset to train two neural network models. The first model performs binary classification to determine whether a traffic record is benign or malicious. If a threat is detected, the second model performs multiclass classification to identify the specific type of attack, such as DoS, DDoS, or PortScan.Both models are fully connected feedforward neural networks that use ReLU activation for efficient training and dropout for regularization. The data is preprocessed with label encoding for categorical fields like Protocol and Flags, and standardized using a scaler for numerical consistency. The complete system is deployed through a Streamlit web interface. Users can input traffic data via form or upload CSV files for batch analysis. Results are displayed in an interactive table with options for downloading and visualization. This framework shows how deep learning can enhance the speed, accuracy, and scalability of cyber threat detection and classification.

Keywords: Cybersecurity, Deep Learning, Threat Classification, Neural Networks, Network Traffic Analysis, Binary Classification, Multiclass Classifications, ReLU Activation, Streamlit Application, Cyber Fed Defender.

I. INTRODUCTION

With the rise of interconnected systems and digital transformation across industries, the threat landscape in cybersecurity has become significantly more dynamic and complex. Organizations today— whether in government, finance, healthcare, or defence are frequently targeted by attackers who exploit system vulnerabilities using sophisticated and evolving tactics. Traditional security approaches, such as rule-based intrusion detection systems and signature-based threat detection, often fall short when facing novel, stealthy, or multi-phase attacks. This growing challenge calls for intelligent, adaptive solutions capable of identifying threats in real time.

To meet this need, this project proposes a deep learning-based system designed to classify network threats by analysing structured traffic data. The goal is twofold: first, to determine whether a given

network flow is safe or malicious (binary classification), and second, to identify the specific type of threat if malicious activity is detected (multiclass classification). Threat types such as Denial of Service (DoS), Distributed Denial of Service (DDoS), Port Scanning, and others are handled through this layered detection approach, enabling more accurate and granular insights.

The system is trained using the CyberFedDefender dataset, which contains labelled records of network behaviour. Two fully connected deep neural networks are used: one for threat detection and another for attack type classification. These models leverage ReLU activation functions to introduce non-linearity and dropout regularization to reduce overfitting. Data preparation steps include label encoding of categorical features like protocol and flags, and standardization of numerical attributes to ensure model consistency and training stability.

To enhance usability, the entire framework is deployed through an intuitive Streamlit web application. This interface allows users to either input single traffic records for instant evaluation or upload bulk data via CSV files for batch processing. The application displays predictions, provides visual summaries such as attack distribution charts, and allows users to download results for further analysis.

Ultimately, this system highlights the potential of deep learning in building intelligent and scalable cybersecurity tools. By automating the detection and classification of threats, it not only improves the speed and reliability of cyber defence strategies but also lays the groundwork for more advanced, AI- driven intrusion detection systems in the future.

II. LITERATURE REVIEW

A. *Understanding Threat Classification Systems*

Threat classification forms a foundational component of modern cybersecurity infrastructure. These systems analyze network traffic to determine whether an activity is normal or potentially harmful. More advanced implementations go beyond detection, offering insight into the nature of malicious behavior by categorizing attacks into various types like Distributed Denial of Service (DDoS), Port Scanning, or DoS. By learning from historical patterns, intelligent classification systems improve detection accuracy and provide timely alerts for defensive action.

B. *Traditional Intrusion Detection Methods*

Earlier cybersecurity tools depended on fixed signatures and rule-based detection. Firewalls, static filters, and pattern-matching techniques were commonly used to flag known threats. These systems worked well for repeated, recognizable attacks but were ineffective in detecting new or adaptive intrusions. Moreover, they often generated high false alarm rates due to their rigid configurations and lack of learning capability**.**

C. *Role of Deep Learning in Threat Classification*

Deep learning introduces a data-driven approach that can identify intricate relationships within network activity. Fully connected neural networks, equipped with nonlinear activation functions like ReLU and techniques such as dropout, are capable of modelling complex behaviors in high-dimensional input data. These models automatically learn features during training, eliminating the need for extensive manual preprocessing and offering improved performance in classifying both common and sophisticated threats.

D. *Machine Learning and AI in Cyber Defense*

Machine learning has become a vital tool in cyber defense, offering methods that learn from historical data to predict threats. Algorithms such as Random Forests, Decision Trees, and Support Vector Machines have been widely used to detect abnormal patterns in network traffic. Deep learning models, on the other hand, handle large volumes of data more efficiently and adapt better to evolving threat landscapes by learning directly from raw or normalized inputs.

E. *Review of Related Work*

Prior research in cybersecurity has explored various detection methods, which can be grouped into traditional machine learning, deep learning, and hybrid models. Widely accepted datasets such as NSL-KDD, CICIDS2017, and CyberFedDefender have served as benchmarks in many studies. Recent advancements show that deep learning approaches often outperform conventional models, particularly in detecting unknown or zero-day attacks. Multi- stage classification systems, where threats are first detected and then classified, have proven effective in improving modularity and accuracy.

F. *Contribution of Current Work*

This project introduces a two-tiered deep learning model for identifying and classifying threats using structured data from the CyberFedDefender dataset. The first model performs binary classification to distinguish threats from normal traffic, while the second categorizes the threats into specific types. The system is wrapped into a Streamlit-based application that allows both individual and batch processing, visual feedback, and downloadable results. The design prioritizes ease of use, modularity, and accuracy, making it a practical tool for academic environments and real-world cybersecurity workflows**.**

III. EXISTING SYSTEM AND LIMITATIONS

A. *Overview of Existing Threat Detection Systems*

Traditional threat detection systems have long served as the foundation of network security. These systems include rule-based firewalls, intrusion detection systems (IDS), and antivirus software. When a signature matches incoming data, an alert is triggered or a connection is blocked. While these systems are effective for detecting previously identified threats, they are limited in their ability to respond to unfamiliar or sophisticated attacks, such as Advanced Persistent Threats (APTs). More recent advancements have introduced machine learning models into cybersecurity tools. These systems use statistical methods to identify deviations from normal behavior, flagging them as potential threats. Some commercial security platforms now combine behavioural analysis with traditional signature detection. However, many of these systems still lack the flexibility and learning capacity required to keep up with evolving attack techniques. They also tend to require ongoing human oversight, manual rule updates, and significant tuning to remain effective.

B. Functional Scope of Traditional Systems

Traditional cybersecurity solutions offer basic threat monitoring and alerting capabilities. Their primary components include: Signature scanning based on known attack patterns Packet inspection and protocol analysis Fixed threshold alerts for abnormal data volumes or connections Manual rule creation and filtering Event logging and security notifications These systems typically work well in controlled or predictable environments, detecting common threats with low resource consumption. They are often integrated into organizational networks as baseline protection tools or used in academic contexts for introductory cybersecurity education.

C. Limitations of Existing Lightweight Systems

Despite their foundational importance, traditional and lightweight cybersecurity systems have notable limitations: Static Logic: They do not adapt to new or unknown attack patterns, limiting their effectiveness against evolving threats. Frequent Maintenance: Updating signature libraries and rules requires constant manual effort, which can lead to delays in detecting newer attacks. Poor Granularity: Most systems detect a threat but fail to provide specific categorization, making it harder to understand the nature or severity of the attack. High False Positives: Legitimate traffic can be incorrectly flagged as malicious, leading to unnecessary alerts and reduced trust in the system. Lack of Learning Capability: These systems cannot learn from previous incidents or improve their accuracy over time.

D. Academic Utility vs Practical Deployment

While traditional threat detection methods are still relevant for basic monitoring and educational purposes, they lack the adaptability and intelligence required for real-world deployment. In practical environments, especially those handling large-scale or sensitive data, a more dynamic and intelligent solution is necessary. The system proposed in this project addresses these shortcomings by introducing a two-stage deep learning approach: one model detects whether a threat is present, and a second model classifies the type of threat. This design improves accuracy, provides actionable insights, and reduces the need for manual intervention. With an interactive user interface, the system is also more accessible and scalable, making it suitable for both research and real-world cybersecurity operations.

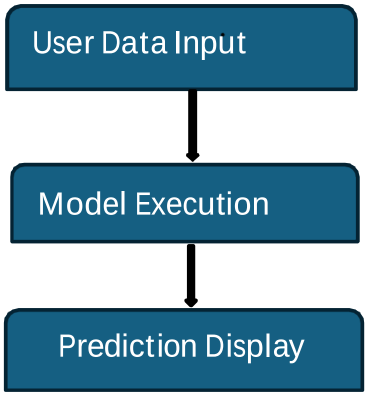
IV. PROPOSED METHODOLOGY

A. Interactive Interface Using Streamlit

The threat classification system starts with a simple and accessible interface developed using **Streamlit**, a Python-based framework for building web applications. This interface provides the following functionalities:

- Uploading a CSV file containing multiple network records
- Entering a single record manually for instant prediction
- Displaying prediction results along with download options

The interface is intuitive and designed to accommodate both experienced users and beginners, allowing for quick interaction with the underlying machine learning models.

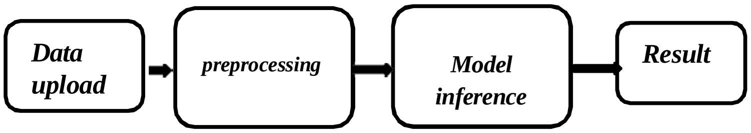


B. Backend Workflow Management

Once the input data is received through the interface, the backend processes it using a predefined logic. This includes:

- Reading and verifying CSV input files
- Performing preprocessing tasks like standardization of numeric features and encoding of categorical fields (e.g., protocol, flags)
- Routing the preprocessed data to the correct deep learning model
- Generating and displaying predictions along with confidence scores

The backend ensures data quality, consistency, and compatibility before any classification is performed.



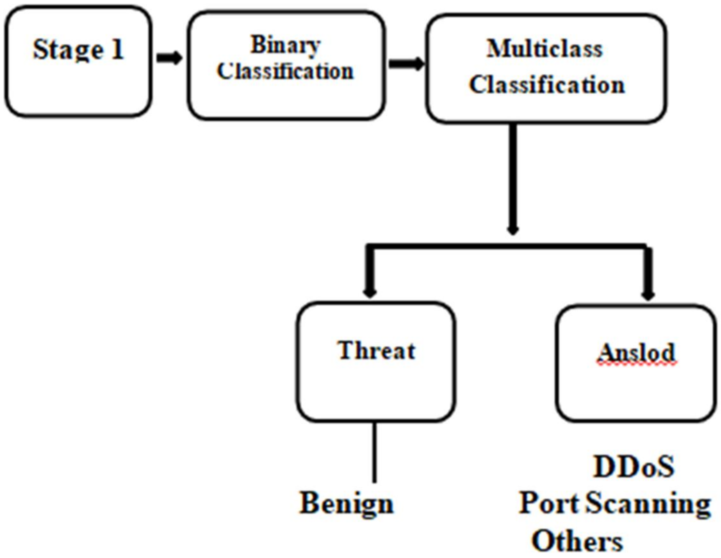
C. Deep Learning Classification Pipeline

At the core of the system is a two-stage deep learning classification engine:

1) Stage 1: Binary Threat Detection This model decides whether the network traffic instance is normal or malicious. It uses a feedforward neural network architecture with ReLU activation and dropout regularization.
2) Stage 2: Threat Type Identification If the input is classified as malicious, a second model is triggered to determine the specific type of threat, such as:

o Denial of Service (DoS)
o Distributed Denial of Service (DDoS)
o Port Scanning
o Other attack categories

Both models use features extracted from structured traffic data, including flow rates, byte counts, protocol types, and port numbers. These are processed using previously trained encoders and scalers to match the model training input format.

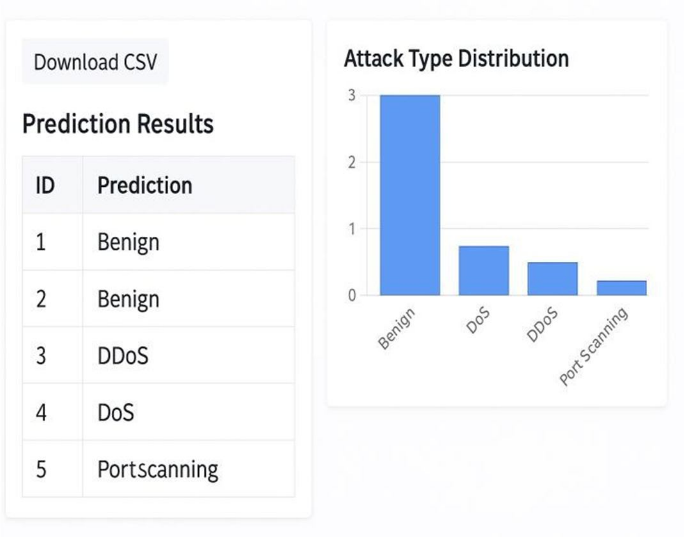


D. Prediction Output and Visualization

Following the classification process, the system generates user-friendly outputs such as:

- A table showing the predicted threat class or benign status
- Downloadable results in CSV format for batch predictions
- Bar graphs and visual summaries showing the distribution of threat types detected across the dataset

This end-to-end feedback loop allows users to not only get predictions but also understand the breakdown of results and export them for reporting or further analysis.



Download CSV

Prediction Results

ID	Prediction
1	Benign
2	Benign
3	DDoS
4	DoS
5	Portscanning

Attack Type Distribution

V. DESIGN METHODOLOGY

A. Method Overview

The system is built around a modular architecture designed to process structured network traffic data for automated threat detection and classification. Users begin by submitting traffic records through a web interface built using Streamlit. This can be done either by uploading a CSV file or manually entering a single record.

Once the data is submitted, the backend performs a series of preprocessing tasks. These include converting categorical fields such as protocol or flag types into numerical form using label encoders and scaling continuous values like byte and packet counts for uniformity. After preprocessing, the data flows through a two-stage deep learning pipeline.

In the first stage, the system uses a binary classification model to determine if the traffic is malicious. If the record is considered a threat, it is passed to a second model, which identifies the specific category of the threat—such as a DoS attack or port scan. The results are then displayed in the web interface, both in tabular form and as graphical summaries. The design ensures that each module works independently, making the system easy to maintain and extend.

Algorithm Used
Name of Algorithm: Two-Stage Deep Learning- Based Threat Classification
Workflow:

1. Accept network data from the user (via manual input or CSV upload).
2. For each record:
   - Apply preprocessing (scaling and label encoding).
   - Use the first model to detect whether the traffic is benign or malicious.
   - If malicious, use a second model to classify the specific attack type.
3. Show predictions directly in the interface.
4. Allow the user to download results and visualize overall threat distribution.

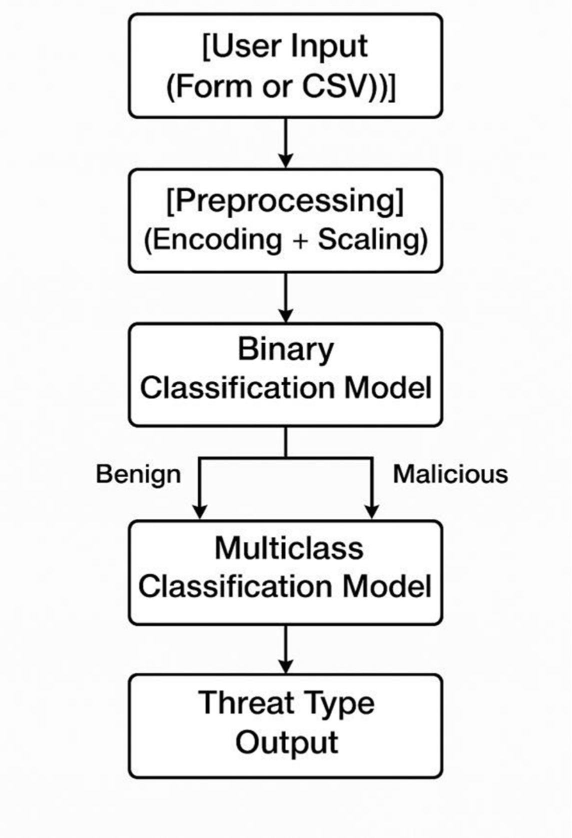
B. Pseudocode

Python code:

```
function classify_network_data(record): processed = preprocess(record)
threat_status = binary_model. predict(processed)
if threat_status == "Benign": return "Benign"
else:
threat_type = multiclass_model.predict(processed) return threat_type
```

C. Conceptual Flow Diagram

You can represent your system visually with this flow:



This diagram outlines the flow from input to prediction and helps highlight the separation of detection and classification stages.

D. Modularity and Extensibility

This system is designed to allow future upgrades without major architectural changes. Possible extensions include:

1) Replacing the current deep neural network models with more complex ones like CNNs or Transformers for better accuracy.
2) Integrating with real-time packet capture tools to enable live monitoring instead of static file uploads.
3) Adding feedback mechanisms to retrain and improve model performance with new data over time.
4) Deploying the system on embedded hardware for use in environments with limited computing resources.
5) Linking with security tools like firewalls or alerting systems to trigger automated responses when threats are detected.

This flexible design ensures that the system is not only effective today but can evolve as cybersecurity challenges become more complex.

VI. IMPLEMENTATION

This project involves building a deep learning-based classification system to detect and categorize cyber threats using structured network traffic data. The solution is designed to be user-friendly, efficient, and scalable, combining trained deep learning models with an interactive web interface. The implementation is done using Python due to its powerful ecosystem of libraries for machine learning and application development.

The system operates in two stages:

a. Binary classification determines whether a traffic record is benign or malicious.
b. Multiclass classification assigns a specific label to malicious records (e.g., DoS, DDoS, Port Scan).

The web interface, created using Streamlit, allows users to interact with the system by uploading traffic data or entering input manually. Results are displayed in a clear, interactive format with support for visual summaries and downloadable reports.

A. Software Environment

- Programming Language: Python 3.8+
- Libraries Used:
  - o TensorFlow / Keras – to build and load deep learning models
  - o NumPy & Pandas – for numerical operations and data handling
  - o Scikit-learn – for feature encoding and scaling
  - o Streamlit – for creating the interactive web interface
  - o Matplotlib / Plotly – for generating charts and visualizations
- Operating System: Compatible with Windows, Linux, and macOS

B. Project Structure

The implementation is divided into three main components to ensure clarity, modularity, and ease of future expansion.

1) Model Inference Logic (Backend)

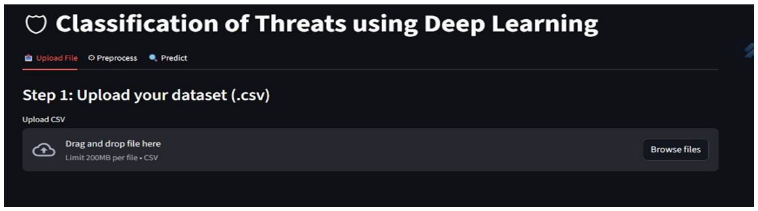
This component handles the classification logic using two pre-trained neural network models saved in .h5 format. The pipeline includes:

- Loading the input features and preprocessing them with pre-saved encoders and scalers
- Running the binary model to detect if a record is benign or a threat
- If classified as a threat, running the second model to determine the type of attack
- Returning the final prediction for each record The neural networks are dense feedforward models with ReLU activation and dropout layers for regularization.

2) Web Interface (Streamlit App)

The web application allows users to interact with the models through a simple and clean interface. Key features include:

- Uploading a CSV file with multiple records or entering a single input via form fields
- Displaying predictions directly in the browser in tabular form
- Providing a downloadable CSV file of all predictions

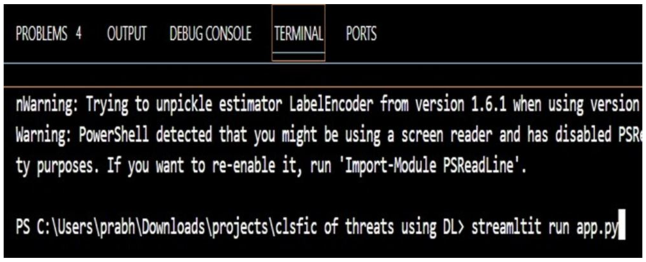


Classification of Threats using Deep Learning

Step 1: Upload your dataset (.csv)

Upload CSV

Drag and drop file here
Limit 200MB per file • CSV



```
nWarning: Trying to unpickle estimator LabelEncoder from version 1.6.1 when using version
Warning: PowerShell detected that you might be using a screen reader and has disabled PSRe
ty purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\prabh\Downloads\projects\clsfic of threats using DL> streamltit run app.py
```

- Visualizing attack type distribution using bar charts

The interface makes the system accessible to users with varying technical backgrounds.

3) Preprocessing Assets

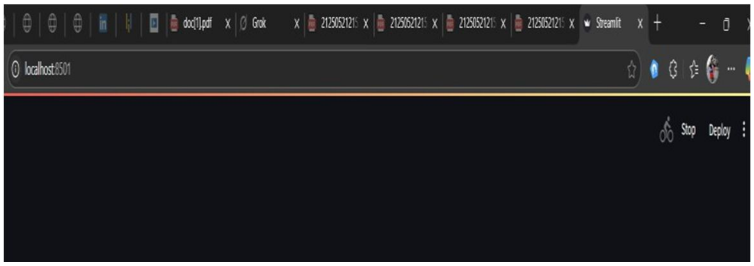
To ensure consistent input for the models, the following resources are loaded during runtime:

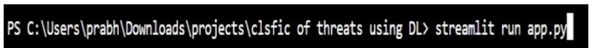
- Scaler – for normalizing numerical values
- Label encoders – for converting categorical fields to numeric format
- Feature names – for maintaining correct input order

These assets were created during the model training phase and are reused during inference.

C. *Initial Interface View*

Upon launching the application, users see a welcome screen with project details and instructions. They can either upload a dataset or enter features manually for prediction. This layout provides a clear starting point for interacting with the system.

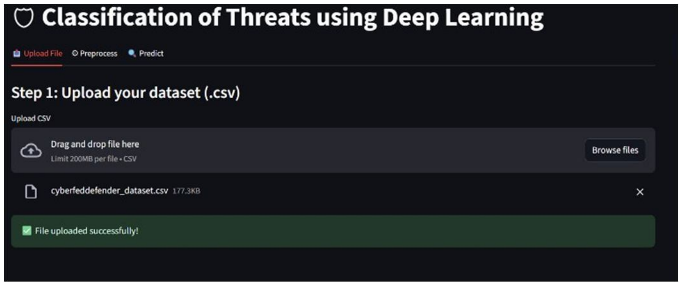




```
PS C:\Users\prabh\Downloads\projects\clsfic of threats using DL> streamlit run app.py
```

D. *File Upload and Preview*

Once a CSV file is uploaded, the app displays the contents for review. This ensures users can verify their data before running predictions. The application also checks for format compatibility before proceeding.



🛡 Classification of Threats using Deep Learning

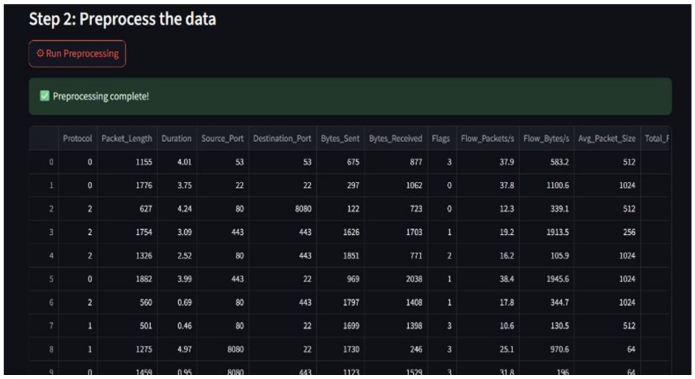
Step 1: Upload your dataset (.csv)

Upload CSV

Drag and drop file here
Limit 200MB per file • CSV

cyberfeddefender_dataset.csv 177.3KB

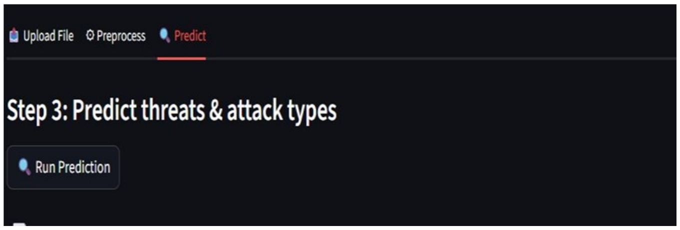
✓ File uploaded successfully!



Step 2: Preprocess the data

✓ Preprocessing complete!

	Protocol	Packet_Length	Duration	Source_Port	Destination_Port	Bytes_Sent	Bytes_Received	Flags	Flow_Packets/s	Flow_Bytes/s	Avg_Packet_Size	Total_F
0	0	1155	4.01	53	53	675	877	3	37.9	583.2	512	
1	0	1776	3.75	22	22	297	1062	0	37.8	1100.6	1024	
2	2	627	4.24	80	8080	122	723	0	12.3	339.1	512	
3	2	1754	3.09	443	443	1626	1703	1	19.2	1913.5	256	
4	2	1326	2.52	80	443	1851	771	2	16.2	105.9	1024	
5	0	1882	3.99	443	22	969	2038	1	38.4	1945.6	1024	
6	2	560	0.69	80	443	1797	1408	1	17.8	344.7	1024	
7	1	501	0.46	80	22	1699	1398	3	10.6	130.5	512	
8	1	1275	4.97	8080	22	1730	246	3	25.1	970.6	64	
9	0	1459	0.95	8080	443	1123	1529	3	31.8	196	64	



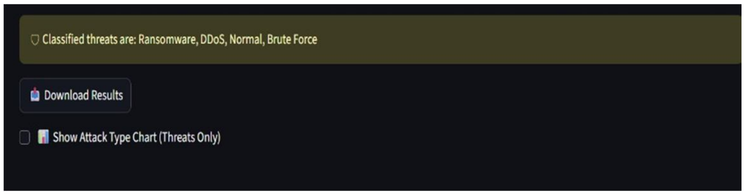
Step 3: Predict threats & attack types

E. *Prediction Output and Feedback*

After processing, results are shown in the following formats:

- Data Table – showing each record’s prediction (e.g., benign, DoS, DDoS)
- Download Option – allowing users to export results in CSV format
- Bar Graph – summarizing the count of different types of threats detected

This feedback loop enables users to interpret model predictions effectively and supports further security analysis.



🛡 Classified threats are: Ransomware, DDoS, Normal, Brute Force

VII. CONCLUSION

This project showcases the use of deep learning to detect and classify cyber threats from network traffic data. The two-stage model helps first identify whether the traffic is safe or harmful and then further classifies harmful traffic into specific attack types. A simple web interface allows users to upload data, view predictions, and download results. The system is efficient, easy to use, and provides a strong foundation for building advanced, real-time threat detection tools in cybersecurity.

REFERENCES

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. A foundational textbook explaining deep learning concepts, architectures, and applications, including classification.

[2] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539 Reviews major advancements in deep learning, including applications to classification problems.

[3] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:1408.5882. While focused on text, it provides insights into how DNNs classify complex data, similar to network traffic.

[4] Wu, Z., & Xie, Y. (2020). Cyber Threat Detection using Deep Learning Techniques: A Survey. Journal of Cybersecurity and Privacy, 1(1), 1–25. Overview of how deep learning is used in threat detection and classification.

[5] Al-Rimy, B. A. S., Maarof, M. A., & Shaid, S. Z. M. (2018). Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. Computers & Security, 74, 144–166. https://doi.org/10.1016/j.cose.2018.01.001 Highlights various cyber threats relevant for classification models.

[6] Liu, H., Lang, B., Liu, M., & Yan, H. (2019). CNN and RNN based payload classification methods for attack detection. Knowledge- Based Systems, 163, 332–341. Discusses deep learning models (CNN/RNN) for threat classification in network traffic.

[7] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Applying deep learning approaches for network traffic classification and intrusion detection. Procedia Computer Science, 132, 1016–1023. Provides a comparative look at using deep neural networks in cybersecurity contexts.

[8] Moustafa, N., & Slay, J. (2015). UNSW- NB15: A comprehensive data set for network intrusion detection systems. Military Communications and Information Systems Conference, IEEE.Describes a dataset similar to CyberFedDefender used for evaluating intrusion detection models.

[9] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41–50. A widely cited paper showing DNNs applied for real-time intrusion detection.

[10] Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A deep learning approach for network intrusion detection system. Proceedings of the 9th EAI International Conference on Bio- inspired Information and Communications Technologies. Introduces a deep architecture for binary and multiclass intrusion classification.

[11] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: methods, systems and tools. IEEE Communications Surveys & Tutorials, 16(1), 303–336.Survey of anomaly-based and signature-based detection systems.

[12] Lopez-Martin, M., Carro, B., Sanchez- Esguevillas, A., & Lloret, J. (2017). Network traffic classifier with convolutional and recurrent neural networks for Internet of Things. IEEE Access, 5, 18042–18050. – Focuses on traffic classification in modern IoT networks using deep models.