**A Synopsis Report on**

Zero Day Malware Detection Using Deep Learning Model

**Submitted in partial fulfillment for the award of the degree of**

**Bachelor of Technology in**

**Computer Science Engineering (Artificial Intelligence & Data Science)**

**Submitted By**

|  |  |
| --- | --- |
| Vibhor | 2821952 |
| Svani | 2820353 |
| Vanshika Wadhwa | 2820361 |
| Kunal | 2821956 |

**Under the Supervision of**

## MS. JYOTI

ASSISTANT PROFESSOR



**Panipat Institute of Engineering & Technology, Samalkha, Panipat**

**Affiliated to**



**Kurukshetra University Kurukshetra, India (2023-2024)**

## Project Credentials

|  |  |  |
| --- | --- | --- |
| Project title | **ZERO DAY MALWARE DETECTION USING DEEP LEARNING MODEL** |  |
| Project Domain/Area |  |
| Group members | Svani 2820353  Vanshika Wadhwa 2820361  Vibhor 2821952  Kunal 2821956 |  |
| Group Id  *(to be allotted by project coordinator)* | G-11 |  |
| **Supervisor’s Name** | MS. JYOTI |  |
| **Supervisor’s Designation** | ASSISTANT PROFESSOR |  |

Supervisor’s Consent

|  |  |
| --- | --- |
| The synopsis of final year project work titled……………………………………………………………………………  by the students’ group id. has been written with my consent and every section  of this synopsis report is reflecting the work to be carried out by the group. | *(Signature of supervisor with date)* |

**Department Project Evaluation Committee (DPEC) Remarks**

The project is by DPEC. The group is advised to submit progress of the

project work in progress presentation-1 to be held on……………………………………………….

OR

The project is by DPEC. The group is advised to submit the synopsis report

again after making changes as suggested by DPEC on …………………........................................

Name & Signature of DPEC member (s) with date

# OBJECTIVE

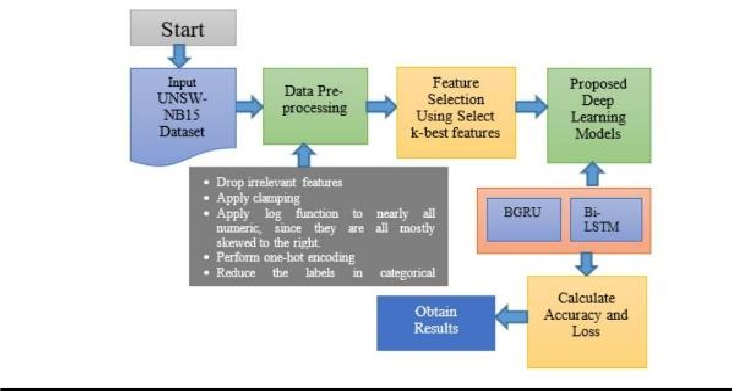


The primary objective of utilizing deep learning models like GRU and LSTM for detecting zero-day malware is to improve the capability of detecting and defending against novel malware that cannot be identified by traditional antivirus solutions due

to their lack of predefined signatures or patterns.

* To improve detection of security attack by using Instruction Detection System (IDS) as a features extract for input into a classification algorithm.
* To effectively use ML from a model pre-trained on security attacks.
* To develop on IDS system that provides the greatest benefit to detect accuracy.

# METHODOLOGY



**Problem Formulation:**Define the problem clearly: Develop a model to accurately detect malware in real-time

**Data Collection:** Gather a diverse dataset of malware samples from different sources,

We will take UNSW-NB15 dataset which is network intrusion dataset consist of 9 attacks which is Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

**Data Preprocessing:** Extract relevant features from the raw data, such as opcode sequences, byte-level n-grams, API calls, and metadata. Normalize the extracted features to a consistent format suitable for deep learning models.

**Model Selection:** Choose appropriate deep learning architectures for malware detection, such as: Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks: Suitable for sequential data like API call sequences. Hybrid architectures combining CNNs and RNNs for capturing both spatial and sequential patterns.

**Model Training:**Split the dataset into training, validation, and test sets. Train the deep learning model on the training data using appropriate loss functions (e.g., binary cross- entropy) and optimization algorithms (e.g., Adam).

**Model Evaluation**:Evaluate the trained model on the validation set using metrics like accuracy, precision, recall, F1-score, and area under the ROC curve (AUC).Fine-tune the model based on validation performance, adjusting hyperparameters or architecture if necessary.

# HARDWARE AND SOFTWARE TO BE USED

### HARDWARE:

**Central Processing Unit (CPU):** A powerful multi-core CPU is necessary for preprocessing data, model training, and inference. Modern CPUs with high clock speeds and multiple cores can significantly speed up computation tasks.

**Graphics Processing Unit (GPU):** GPUs excel in parallel processing tasks and are essential for accelerating deep learning model training

**Memory (RAM):** Sufficient RAM capacity is necessary to store and manipulate large datasets during preprocessing and model training. The amount of RAM required depends on the size of the dataset and the complexity of the deep learning models.

**Storage:** Fast and reliable storage is essential for storing datasets, model parameters, and

### SOFTWARE:

**Deep Learning Frameworks**: Choose deep learning frameworks that provide efficient implementations of popular neural network architectures and optimization algorithms. Some commonly used frameworks include TensorFlow, PyTorch, Keras, and MXNet.

**Development Environment:** Set up a development environment with appropriate tools for coding, debugging, and version control. Integrated development environments (IDEs) such as PyCharm, Visual Studio Code, or Jupyter Notebooks are commonly used.

**Data Preprocessing Tools:** Utilize libraries and tools for data preprocessing tasks such as feature extraction, transformation, and normalization. Libraries like NumPy, Pandas, and Scikit-learn are commonly used for data manipulation and preprocessing.

**Visualization Tools:** Visualize data distributions, model architectures, training curves, and evaluation metrics using plotting libraries like Matplotlib, Seaborn, or Plotly.

**Security Tools:**Integrate security tools and libraries for data encryption, secure communication, access control, and threat detection to protect sensitive data and prevent security breaches.

# LIMITATIONS

**Data Availability and Quality**: Deep learning models require large volumes of high-quality data for effective training. However, obtaining labeled datasets of zero-day malware samples can be challenging due to the scarcity of such instances.

**Imbalanced Data**: In cybersecurity, the distribution of malware samples is highly imbalanced, with a vast majority of benign instances compared to malicious ones.

I**nterpretability**: Deep learning models are often criticized for their lack of interpretability, making it challenging to understand the underlying reasons behind model predictions.Balancing model complexity with interpretability remains a significant challenge in designing effective deep learning-based malware detection systems.

**Generalization to New Threats**: While deep learning models excel at learning patterns from data, their ability to generalize to previously unseen threats is not guaranteed. Zero-day malware by definition has never been encountered before, posing a challenge for detection systems that rely on historical data.

Addressing these limitations requires a multidisciplinary approach that integrates advances in deep learning techniques with domain expertise in cybersecurity. Moreover, collaboration between researchers, industry practitioners, and cybersecurity professionals is essential for advancing the state-of-the-art in zero-day malware detection and mitigating emerging threats effectively.

# FUTURE SCOPE

The future scope of using deep learning models like GRU and LSTM for detecting zero-day malware is promising. As the cybersecurity landscape continues to evolve, deep learning will play an increasingly vital role in safeguarding against zero-day malware threats.

**Feature Extraction**: Deep learning algorithms excel at automatically extracting relevant features from raw data. By analyzing various aspects of malware behavior and characteristics, deep learning models can identify subtle indicators that might not be apparent to traditional detection methods.

**Detection of unknown threats**: Deep learning models can analyze vast amounts of data, including file structures, network traffic, and system behavior, to identify anomalous patterns indicative of zero-day malware

**Scalability**: With the continuous evolution of malware variants and attack techniques, the scalability of detection mechanisms becomes crucial. Deep learning models can be trained on large datasets, allowing them to scale efficiently and adapt to emerging threats without requiring manual intervention.

**Real time detection**: The speed at which zero-day malware can propagate necessitates real- time detection mechanisms. Deep learning models can process incoming data streams in near real-time, enabling rapid identification and response to emerging threats before they cause significant damage.

**Behavioral analysis**: Deep learning techniques, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), can capture complex temporal and spatial relationships in data.

However, it's essential to acknowledge some challenges and considerations associated with deploying deep learning-based malware detection systems, including the need for large and diverse datasets, interpretability of model decisions, and computational resource requirements.

# CONCLUSION

In conclusion, leveraging deep learning models for the detection and mitigation of zero-day malware attacks presents a promising avenue for enhancing cybersecurity defenses. Throughout this discussion, we have explored the various aspects and potentialities of utilizing advanced machine learning techniques to combating emerging threats.

Deep learning models offer significant advantages in their ability to learn complex patterns and behaviors from large datasets, enabling them to detect previously unseen malware variants and zero-day attacks. By analyzing diverse features extracted from malware samples and benign software, these models can effectively differentiate between malicious and non- malicious behavior, thereby providing proactive defense mechanisms against evolving threats.

However, it is essential to acknowledge the challenges and limitations inherent in deploying deep learning-based solutions for zero-day malware detection.

In conclusion, while deep learning models are not a panacea for all cybersecurity threats, they represent a valuable tool in the arsenal of cybersecurity practitioners, enabling them to stay ahead of evolving malware threats and better protect digital assets and infrastructure. By embracing innovation, collaboration, and continuous improvement, we can build a more secure and resilient cyber ecosystem capable of mitigating the risks posed by zero-day malware attacks.

# REFERENCES

1. Abri Faranak, Siami-Namini Sima, Khanghah Mahdi Adl, Soltani Fahimeh Mirza, and Namin Akbar Siami. 2019. Can machine/deep learning classifiers detect zero-day malware with high accuracy? In *Proceedings of the IEEE International Conference on*

*Big Data*. IEEE, 3252–3259.

1. Abusnaina Ahmed, Abuhamad Mohammed, Alasmary Hisham, Anwar Afsah, Jang Rhongho, Salem Saeed, Nyang Daehun, and Mohaisen David. 2022. DL-FHMC: Deep learning-based fine-grained hierarchical learning approach for robust malware

classification. *IEEE Trans. Depend. Sec. Comput.* 19, 5 (Sept.2022), 3432–3447.

1. Abusnaina Ahmed, Alasmary Hisham, Abuhamad Mohammed, Salem Saeed, Nyang DaeHun, and Mohaisen Aziz. 2019. Subgraph-based adversarial examples against graph-based IoT malware detection systems. In *Computational Data and Social*

*Networks*, Tagarelli Andrea and Tong Hanghang (Eds.). Springer, 268–281.