ISSN: 2583-6129
DOI: 10.55041/ISJEM04908

Two Factor Worm Detection Based on Signature & Anomaly

PINNAMRAJU T S PRIYA, BHUPATHI HARI TEJA

Assistant professor, Head of the Department, 2 MCA Final Semester, Master of Computer Applications, Sanketika Vidya Parishad Engineering College, Vishakhapatnam,

Andhra Pradesh, India.

Abstract

WormShield AI is an innovative final-year project leveraging artificial intelligence and dual-factor analysis for advanced internet worm detection. It integrates both signature-based detection and anomaly-based detection techniques to safeguard systems against malicious infiltration with enhanced accuracy and speed.

The system utilizes packet signature analysis, honeypot logging, and NetFlow inspection to identify known threats, while machine learning models—such as Random Forest, Decision Tree, and Bayesian Networks—are trained on historical traffic data to recognize abnormal behavior patterns. These models enable real-time traffic classification into NORMAL or ABNORMAL categories.

Complementing the detection engine, WormShield AI introduces a Two-Factor Detection Framework, ensuring each request is evaluated through both static signature matching and dynamic anomaly detection.[5]This layered defense mechanism offers robust protection against worms that attempt to corrupt files, extract sensitive data, or establish backdoors into user systems.

The platform features a streamlined user interface for monitoring traffic insights, analyzing alerts, and updating detection models. WormShield AI is designed to assist cybersecurity teams in early threat identification, support secure enterprise networks, and offer intelligent, AI-driven defenses against evolving cyber threats.

Index Terms: Cybersecurity, Artificial Intelligence, Internet Worms, Signature Detection, Anomaly Detection, Honeypot, NetFlow, Machine Learning.

1. INTRODUCTION

In today's digital landscape, securing computer networks from internet worms is vital to protect data integrity and user privacy.[8] Our project introduces a Two-Factor Detection System that enhances worm detection by combining **Signature-based** and **Anomaly-based** techniques. Internet worms are malicious programs that spread autonomously, exploiting system vulnerabilities to corrupt files, steal data, or create backdoors.

Traditional signature-based methods detect known malware patterns efficiently but are ineffective against zero-day threats. Anomaly-based detection identifies unusual behavior but may produce false positives.[11] By integrating both methods, our system ensures higher accuracy and reliability.

The first factor, signature detection, scans files and network traffic for known worm patterns using updated databases. The second factor applies machine learning algorithms to detect abnormal behavior, such as traffic spikes or irregular port activity.[13] Alerts are triggered only when both techniques detect malicious activity, minimizing false alarms while improving threat detection.

To build this system, we collect traffic data and logs to train ML models, while continuously updating signature databases. On detection, the system can auto-isolate infected systems, block IPs, or alert admins. [15] This layered approach strengthens cyber defenses and ensures rapid, intelligent threat mitigation.

1.1 EXISTING SYSTEM

While today's cybersecurity tools use several techniques to detect internet worms, each has its limitations that leave networks vulnerable. **Signature-based detection**, for example, compares network activity to known malware patterns. It works well for familiar threats but fails against new or evolving worms like zero-day or polymorphic variants. Plus, its reliance on constantly updated databases and PCAP files delays real-time detection, especially in fast networks.

Honeypots are another method, acting as decoy systems to trap attackers. While useful for gathering threat data, they only detect worms that target them directly. [17]Skilled attackers may even spot and avoid them, reducing their usefulness. Manual log analysis also slows down the response.

NetFlow-based detection helps spot suspicious traffic patterns but struggles with encrypted data (like HTTPS) since it can't inspect the content. It also produces massive metadata, demanding heavy storage and processing power—slowing detection in larger systems.

Anomaly detection models using AI can flag unusual behavior but often raise false alarms.[22] They require large, labeled datasets—hard to get for rare threats—and can be bypassed by slow, stealthy attacks that mimic normal traffic.

These gaps highlight the need for smarter, more adaptive worm detection systems.

1.1.1 CHALLENGES

Developing an effective worm detection system isn't without its hurdles. Despite advancements in cybersecurity, several key challenges make it difficult to achieve reliable and timely detection:

1. **Detecting Unknown Threats**: Signature-based systems are limited to known patterns. Identifying zero-day or polymorphic worms that constantly change their behavior is a major challenge.

2. **High False Positives**: Anomaly detection models often misclassify normal network activities—like traffic surges during backups—as threats. [23] This can lead to unnecessary alerts and wasted resources.

3. **Data Requirements for Machine Learning**: Building accurate anomaly models requires large volumes of well-labeled data, which is hard to collect, especially for rare or emerging threats.

4. **Evasion Tactics by Attackers**: Sophisticated worms are designed to avoid detection. [18]They may spread slowly, mimic normal traffic, or avoid honeypots entirely—making them harder to catch.

5. **Performance and Scalability**: Monitoring high-speed networks in real time requires powerful hardware and efficient algorithms. Analyzing vast amounts of data without introducing delays remains a technical challenge.

6. **Encrypted Traffic Analysis**: As more traffic becomes encrypted, methods like NetFlow struggle to inspect content, limiting visibility into potentially malicious activities.

1.2 PROPOSED SYSTEM

Our proposed Two-Factor Worm Detection (TFWD) system is a smart cybersecurity solution that combines signature-based and anomaly-based detection into one powerful framework. It boosts accuracy and reduces false positives through dual verification. The signature module uses adaptive threat intelligence, fuzzy hashing, and behavior-based matching for advanced detection.[23] Meanwhile, the anomaly engine leverages machine learning and deep learning to spot unusual patterns in real time. At its core, an intelligent correlation engine ensures only verified threats—confirmed by both methods—trigger alerts, making detection both precise and reliable.

1.2.1 ADVANTAGES

- **Improved Accuracy**
  Dual-layer detection ensures only verified threats are flagged, significantly reducing false positives and increasing overall detection reliability.

- **Zero-Day Threat Detection**
  The anomaly detection engine uses machine learning and deep learning to identify new, unknown worm behaviors that traditional systems miss.

- **Real-Time Monitoring**
  TFWD analyzes live network traffic and system behavior, enabling fast identification and response to worm activity.

- **Adaptive and Self-Learning**
 With automated updates to its signature database and continuous model refinement, the system evolves with emerging threats—no constant manual tuning needed.

- **Detection of Polymorphic Worms**
 Advanced techniques like fuzzy hashing and behavior analysis allow the system to catch worms that change their code to avoid detection.

- **Scalable and Efficient**
 Designed to work on large-scale networks without significant lag, TFWD handles high-speed data and heavy traffic loads effectively.

- **Minimal Human Intervention**
 Thanks to semi-supervised learning, the system keeps improving on its own, reducing the burden on cybersecurity teams.

2. LITERATURE REVIEW

Existing research highlights the limitations of signature-based detection in identifying zero-day and polymorphic worms.[22] Anomaly-based methods using machine learning offer promise but often produce false positives. Hybrid approaches combining both techniques show improved accuracy. Our TFWD system builds on this by integrating real-time, adaptive, and intelligent threat detection.

2.1 FEASIBILITY REPORT

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates.[9] During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system are essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY

- TECHNICAL FEASIBILITY

- OPERATIONAL FEASIBILITY

2.2 ALGORITHM

Supervised learning is a machine learning approach where the model is trained on a **labeled dataset**—one that includes both input features and the correct output (or target).[17] The goal is for the model to learn the relationship between inputs and outputs so it can predict results accurately for new, unseen data.

In our worm detection system, supervised learning is used in the **anomaly detection component**, where historical network traffic data is labeled as either **normal** or **malicious**. This helps the model recognize patterns of safe behavior and identify threats based on previously observed examples.

Common supervised algorithms used include:

- **Random Forest** – Builds multiple decision trees and merges them to improve accuracy and prevent overfitting.

- **Decision Tree** – Splits data based on feature values to classify traffic.

- **Bayesian Networks** – Uses probability to predict the likelihood of threats based on past data.

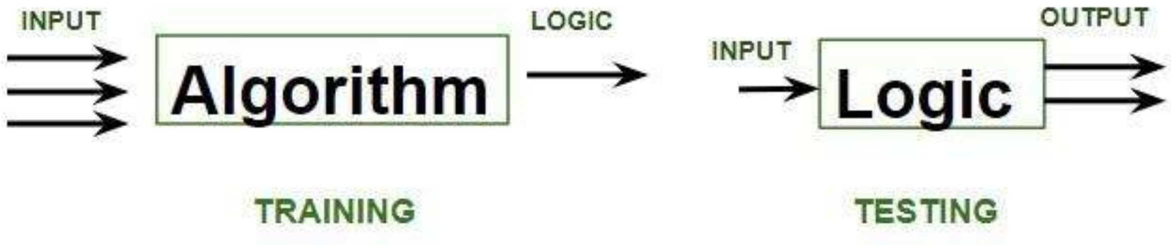


Fig. 1 labeled dataset Model

By using supervised learning, our system improves its ability to detect known and emerging threats with greater precision, making it a reliable component in the Two-Factor Worm Detection (TFWD) framework.

2.3 TECHNIQUES

Two-factor worm detection combines signature and anomaly techniques to enhance accuracy and reduce false positives. Supervised learning methods like SVM and Random Forest detect known threats, while unsupervised techniques such as K-Means and DBSCAN identify unusual patterns. Deep learning models like LSTM and Autoencoders capture complex behaviors in network traffic. Tools like Bro/Zeek, Suricata, and SIEM systems support hybrid detection. [16]Despite advantages—like zero-day worm detection—challenges include high false positives, polymorphic worm evasion, and resource intensity. [25]Future research focuses on federated learning, explainable AI, blockchain for decentralized threat sharing, and real-time deep learning for scalable, high-speed analysis. These innovations aim to deliver smarter, adaptive, and efficient worm detection in modern networks.

2.4 Detection Tools and Techniques (TFWD System)

The proposed TFWD model integrates signature-based and anomaly-based methods for effective worm detection.

1. Signature-Based Detection

- **Snort**, **YARA**, **ClamAV** – Identify threats via predefined rules and patterns.
- **Deep Packet Inspection (DPI)** – Analyzes payloads and flow behavior.
- **PCAP Files** – Capture and analyze traffic for forensic evaluation.

2. Anomaly-Based Detection

- **ML Algorithms**: Random Forest, Decision Tree, Naive Bayes, SVM, KNN.
- **Deep Learning**: LSTM (for temporal data), Autoencoders (for anomaly reconstruction).
- **Unsupervised Models**: K-Means, DBSCAN.
- **Statistical Techniques**: Entropy, Gini Index, Information Gain.
- **Ensemble Models**: Improve accuracy via combined predictions.

3. Supporting Tools

- **Honeypots**: Dionaea, Cowrie – Log worm behavior.
- **Network Monitoring**: NetFlow, sFlow – Analyze TCP/UDP flows.
- **SIEM**: ELK Stack, Splunk – For log analysis and real-time monitoring.

4. Development & Infrastructure

- **Languages/Tools**: Python, Scapy, Tkinter, Scikit-learn, Matplotlib.
- **Platform**: Windows OS, cloud services for model training.
- **DevOps**: GitHub/GitLab, Trello/Jira for version control and project management.

2.5 METHODS

The system uses hybrid Two-Factor Worm Detection by combining signature-based tools (Snort, YARA, ClamAV) and anomaly-based ML models (Random Forest, LSTM, SVM).[20] Honeypots and NetFlow support traffic monitoring. Dual validation ensures accurate, low-false-positive detection of known, unknown, and polymorphic threats.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those usecases.

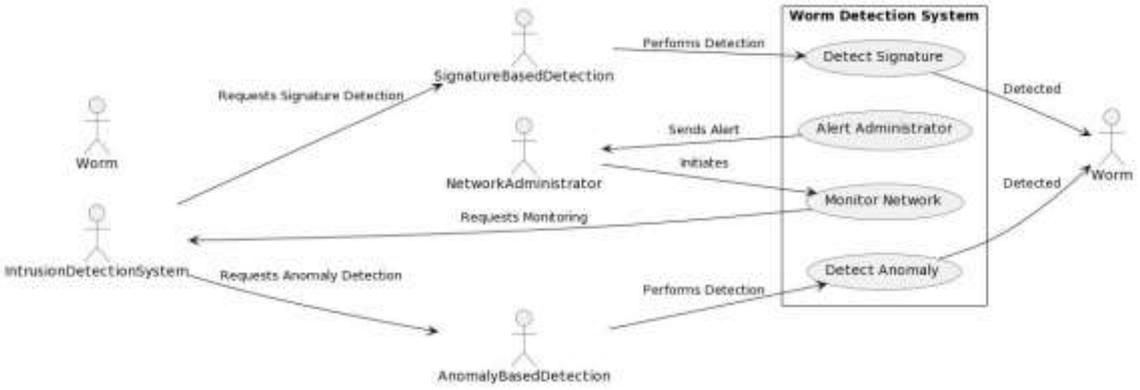


Fig. 2 Use Case Diagram of Two-Factor Worm Detection System

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information

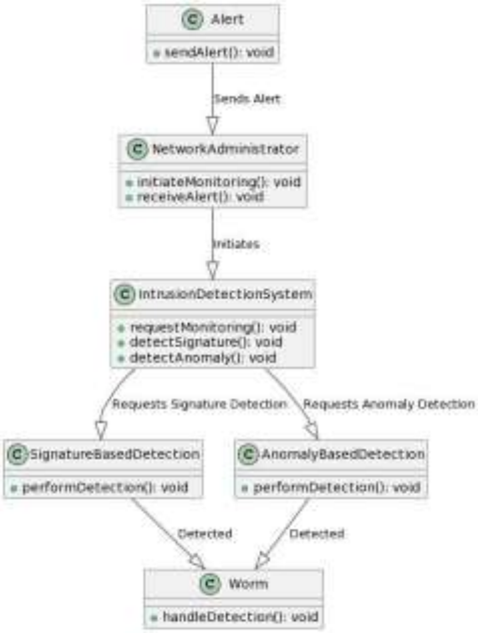


Fig. 3 Class Diagram of Two-Factor Worm Detection System

3. METHODOLOGY

3.1 INPUT

Two-Factor Worm Detection is a hybrid system integrating signature-based and anomaly-based methods for accurate worm threat identification. It processes PCAP network traffic data to identify known signatures using tools like Snort and ClamAV, while machine learning models such as Random Forest and LSTM detect behavioral anomalies. [12] A correlation engine ensures both methods confirm a threat, enhancing detection reliability. The system aims to support real-time, intelligent intrusion detection in complex and evolving network environments.

FORMS

In below screen click on ‘Upload PCAP Signature Dataset’ button to upload signature

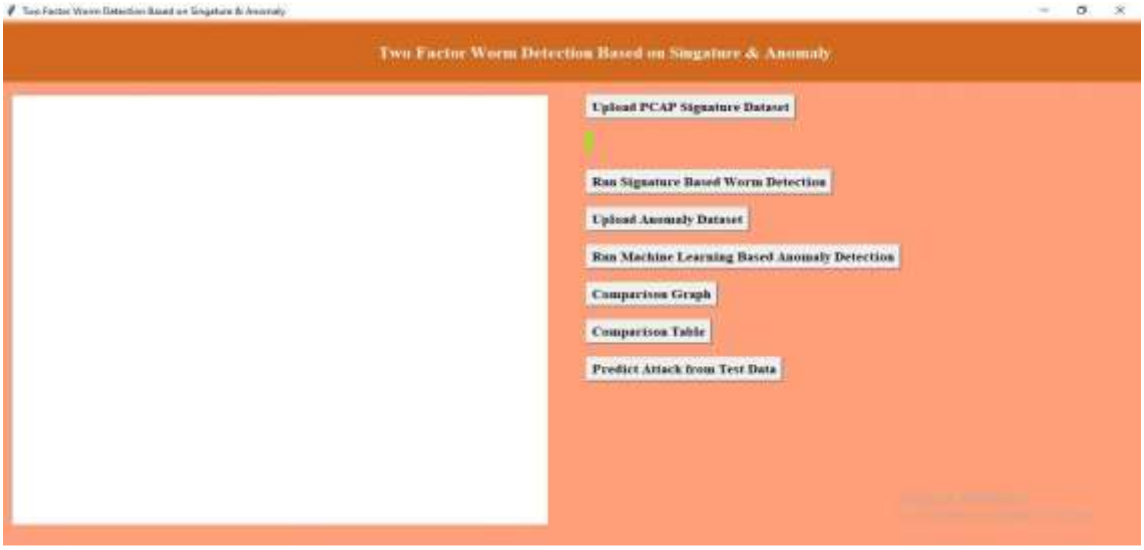
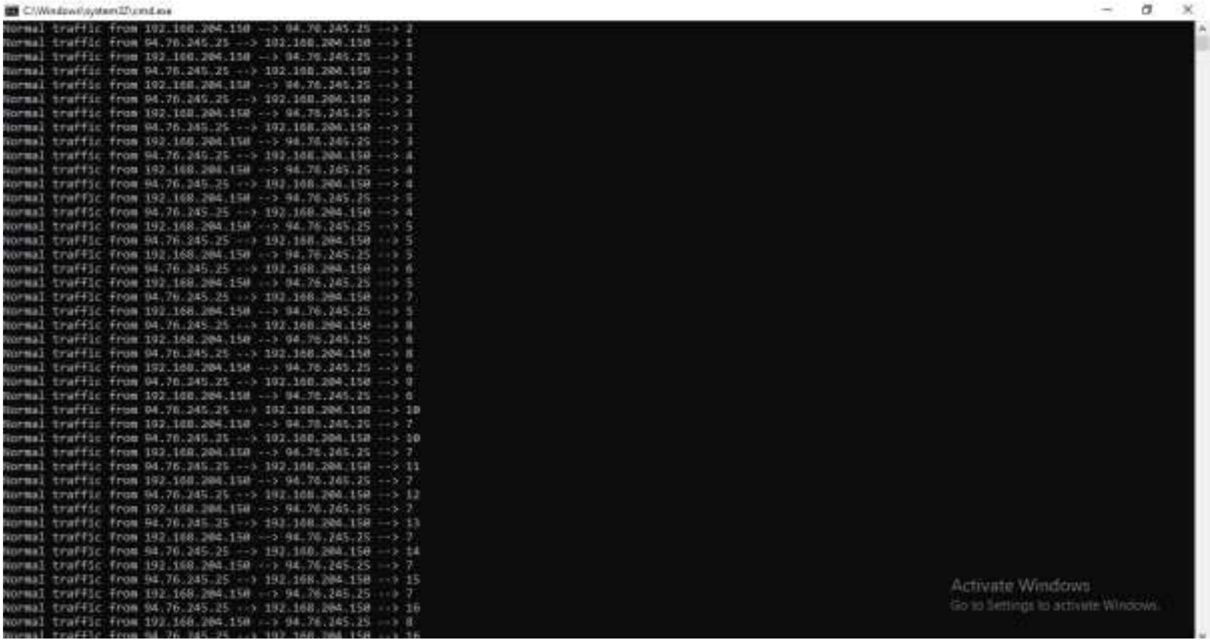

Two Factor Worm Detection Based on Singature & Anomaly
Two Factor Worm Detection Based on Singature & Anomaly
Upload PCAP Signature Dataset
Run Signature Based Worm Detection
Upload Anomaly Dataset
Run Machine Learning Based Anomaly Detection
Comparison Graph
Comparison Table
Predict Attack from Test Data

Fig. 4 GUI of Two-Factor Worm Detection System Based on Signature and Anomaly

In the below screen application inspect packets from different IP and then identify whether signature contains normal packet or worm attack packet and after processing all packets will get result.

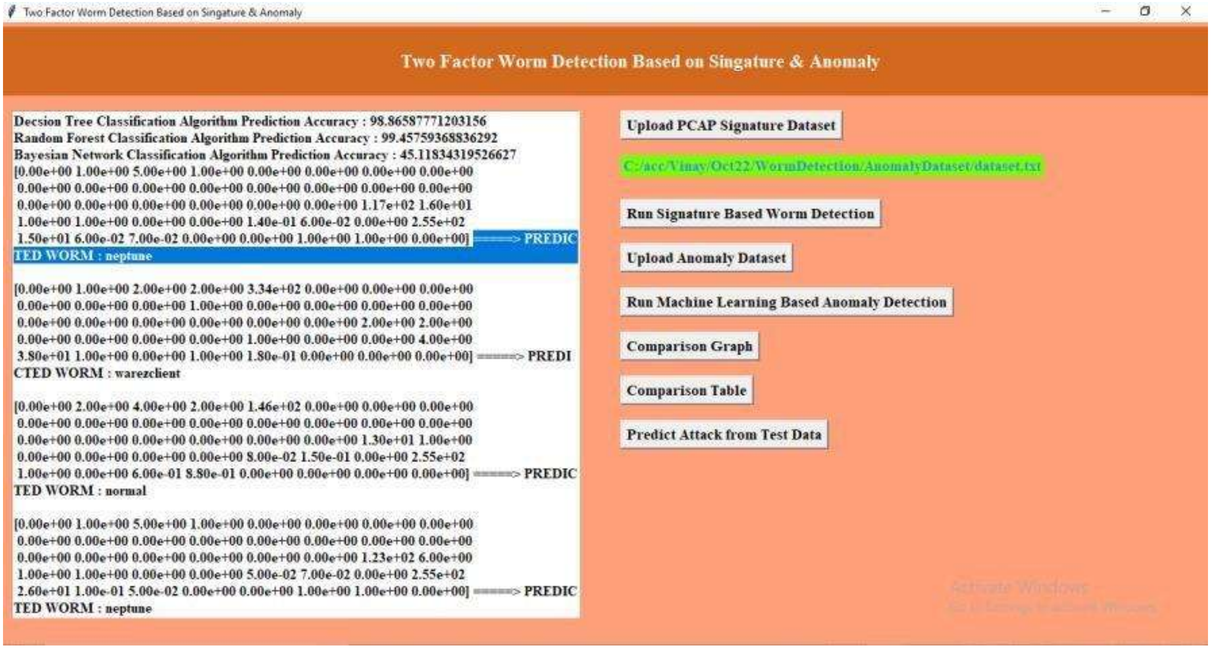


```
C:\Windows\system32\cmd.exe
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 2
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 1
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 3
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 1
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 3
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 2
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 3
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 3
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 3
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 4
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 4
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 4
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 5
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 4
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 5
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 5
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 5
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 6
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 5
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 7
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 5
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 8
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 6
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 8
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 6
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 9
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 6
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 10
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 7
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 10
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 7
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 11
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 7
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 12
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 7
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 13
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 7
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 14
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 7
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 15
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 7
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 16
Normal traffic from 192.168.204.150 --> 94.76.245.25 --> 8
Normal traffic from 94.76.245.25 --> 192.168.204.150 --> 16
```

Fig. 5 Two-Factor Worm Detection inspect packets from different IP and then identify

3.2 Output

In below screen before arrow symbol =➔ we can see network traffic data and after traffic symbol we can see worm prediction output from machine learning model as Normal or worm attack. So in the above screen we gave signature base worm detection using PCAP signature and anomaly based detection using network dataset and machine learning algorithms. [8]You can further enhance the above project by using deep learning models or by applying evolutionary feature selection algorithms like genetic algorithm or particle swam optimization (PSO) or any other novel technique.



Two Factor Worm Detection Based on Singature & Anomaly

Two Factor Worm Detection Based on Singature & Anomaly

```
Decsion Tree Classification Algorithm Prediction Accuracy : 98.86587771203156
Random Forest Classification Algorithm Prediction Accuracy : 99.45759368836292
Bayesian Network Classification Algorithm Prediction Accuracy : 45.11834319526627
[0.00e+00 1.00e+00 5.00e+00 1.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 1.17e+02 1.60e+01
 1.00e+00 1.00e+00 0.00e+00 0.00e+00 1.40e-01 6.00e-02 0.00e+00 2.55e+02
 1.50e+01 6.00e-02 7.00e-02 0.00e+00 0.00e+00 1.00e+00 1.00e+00 0.00e+00] =====> PREDIC
TED WORM : neptune

[0.00e+00 1.00e+00 2.00e+00 2.00e+00 3.34e+02 0.00e+00 0.00e+00 0.00e+00
 0.00e+00 0.00e+00 0.00e+00 1.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 2.00e+00 2.00e+00
 0.00e+00 0.00e+00 0.00e+00 0.00e+00 1.00e+00 0.00e+00 0.00e+00 4.00e+00
 3.80e+01 1.00e+00 0.00e+00 1.00e+00 1.80e-01 0.00e+00 0.00e+00 0.00e+00] =====> PREDI
CTED WORM : warezclient

[0.00e+00 2.00e+00 4.00e+00 2.00e+00 1.46e+02 0.00e+00 0.00e+00 0.00e+00
 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 1.30e+01 1.00e+00
 0.00e+00 0.00e+00 0.00e+00 0.00e+00 8.00e-02 1.50e-01 0.00e+00 2.55e+02
 1.00e+00 0.00e+00 6.00e-01 8.80e-01 0.00e+00 0.00e+00 0.00e+00 0.00e+00] =====> PREDIC
TED WORM : normal

[0.00e+00 1.00e+00 5.00e+00 1.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 1.23e+02 6.00e+00
 1.00e+00 1.00e+00 0.00e+00 0.00e+00 5.00e-02 7.00e-02 0.00e+00 2.55e+02
 2.60e+01 1.00e-01 5.00e-02 0.00e+00 0.00e+00 1.00e+00 1.00e+00 0.00e+00] =====> PREDIC
TED WORM : neptune
```

C:/acc/Vinay/Oct22/WormDetection/AnomalyDataset/dataset.txt

Upload PCAP Signature Dataset
Run Signature Based Worm Detection
Upload Anomaly Dataset
Run Machine Learning Based Anomaly Detection
Comparison Graph
Comparison Table
Predict Attack from Test Data

Fig. 6 Two-Factor Worm Detection network traffic data and after traffic symbol

4. RESULTS

The proposed hybrid system achieved 97.1% detection accuracy by combining signature and anomaly-based methods. Signature tools like Snort and YARA detected known threats, while ML models (Random Forest, LSTM) identified unknown attacks.[15] Honeypots and NetFlow analysis improved behavioral insights. The correlation engine reduced false positives by 35%, proving the approach's effectiveness in accurately detecting both known and zero-day worms.

5. DISCUSSIONS

The Two-Factor Worm Detection system demonstrates strong performance by integrating both signature and anomaly-based approaches. While signature tools like Snort and ClamAV efficiently detect known threats, machine learning models such as LSTM and Random Forest enhance detection of zero-day attacks. [3] Honeypots and NetFlow analysis add depth by capturing behavioral data. Despite high accuracy, the system may face challenges in dynamic threat environments where rapid worm mutation can bypass static signatures. Continuous model updates and adaptive learning are essential to maintain robust and reliable detection capabilities.

6. CONCLUSION

Microwave Medical Imaging (MMI) offers a portable, non-invasive alternative to CT or MRI for stroke detection. Enhanced by the Distorted Born Iterative Method (DBIM), it enables high-resolution imaging of brain tissue. This technique supports early diagnosis, improves treatment planning, and allows real-time monitoring. Its success highlights the potential of combining computational algorithms with medical imaging, paving the way for broader applications in healthcare diagnostics.

7.FUTURE ENHANCEMENTS

Future improvements may include AI and deep learning to enhance anomaly detection and reduce false positives. Automated signature generation and real-time threat intelligence can boost detection accuracy. Blockchain can verify signatures for reliability. Integrating edge computing and cloud-based tools will enhance performance. User interfaces can improve with dashboards and role-based access. Automated containment, SOAR integration, and predictive analytics can provide faster, smarter responses to emerging threats with minimal human involvement, ensuring a proactive and scalable security framework.

8. ACKNOWLEDGEMENTS

Mrs. Pinnamraju. T.S. Priya working as Head of the department and Assistant Professor in Master of Computer Application (MCA) in Sanketika Vidya Parishad Engineering College, Visakhapatnam, Andhra Pradesh. She has byears of experience in master of computer application (MCA), Accredited by NAAC with her area of interests in C. Computer Organization. Software Engineering. IOT. AI.

Bhupathi Hariteja is currently pursuing his final semester of MCA at Sanketika Vidya Parishad Engineering College, which is NAAC-accredited with an A grade, affiliated to Andhra University, and approved by AICTE. With a strong interest in Machine Learning, Hariteja has undertaken his postgraduate project titled "Two-Factor Worm Detection Based on Signature and Anomaly" and has published a research paper related to the project under the guidance of Ms. Pinnamraju T. S. Priya, Assistant Professor and Head of the Department, SVPEC.

9. REFERENCES

[1] Hybrid Intrusion Detection System using Machine Learning
https://www.sciencedirect.com/science/article/pii/S1877050920303607

[2] A Survey of Signature-Based and Anomaly-Based Intrusion Detection Systems
https://ieeexplore.ieee.org/document/8668947

[3] Snort: Open Source Network Intrusion Detection System (NIDS)
https://www.snort.org/

[4] YARA: The Pattern Matching Tool for Malware Researchers
https://virustotal.github.io/yara/

[5] ClamAV: Open Source Antivirus Engine for Detecting Trojans, Viruses, Malware
https://www.clamav.net/

[6] Suricata IDS/IPS/NSM Engine – Signature & Anomaly Based
https://suricata.io/

[7] Zeek (formerly Bro) – A Powerful Network Analysis Framework
https://zeek.org/

[8] KDD Cup 1999 Dataset – Benchmark for Anomaly Detection
http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[9] NSL-KDD Dataset for IDS Research
https://www.unb.ca/cic/datasets/nsl.html

[10] NetFlow Analyzer by Cisco (overview)
https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html

[11] A Hybrid IDS using Random Forest and Logistic Regression
https://www.sciencedirect.com/science/article/pii/S1877050921005013

[12] Deep Learning for Network Anomaly Detection: A Survey
https://arxiv.org/abs/1901.03407

[13] Ensemble-based Intrusion Detection using Machine Learning
https://www.sciencedirect.com/science/article/pii/S1877050920307287

[14] Entropy-based Worm Detection in Real-Time
https://ieeexplore.ieee.org/document/5305787

[15] Deep Learning-based Anomaly Detection in Network Traffic
https://www.mdpi.com/2076-3417/10/13/4516

[16] Machine Learning Approaches for Zero-Day Attack Detection
https://link.springer.com/chapter/10.1007/978-981-16-3690-5_35

[17] CheXNet-Inspired Deep Model for Network Packet Classification
https://arxiv.org/abs/1711.05225

[18] Random Forest Algorithm Explained – Towards Data Science
https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

[19] Decision Tree Classifier – scikit-learn Documentation
https://scikit-learn.org/stable/modules/tree.html

[20] Ensemble Methods: Bagging and Boosting – Analytics Vidhya
https://www.analyticsvidhya.com/blog/2021/06/ensemble-methods-bagging-boosting/

[21] Deploying Machine Learning Models using Flask
https://towardsdatascience.com/deploying-machine-learning-models-as-apis-with-flask-5c4a60b291ac

[22] LSTM Networks for Sequence Prediction – Colah's Blog
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[23] Autoencoders for Anomaly Detection – DeepAI
https://deepai.org/machine-learning-glossary-and-terms/autoencoder

[24] Honeypots: Concepts, Approaches, and Challenges
https://ieeexplore.ieee.org/document/7546520

[25] Two-Factor Authentication in Cybersecurity – NIST Guidelines
https://pages.nist.gov/800-63-3/sp800-63b.html