Grenze International Journal of Engineering and Technology, June Issue

Detection and Classification of Network Attacks in Encrypted Wireless Traffic using Machine Learning Techniques

Puttaraj C Tembadamani[1] and Bhat Geetalaxmi Jairam[2]
[1-2]Department of Information Science and Engineering, The National Institute of Engineering, Mysore, Karnataka, India
Email: puttarajtembadamani@gmail.com, bgj@nie.ac.in

***Abstract*— As encryption becomes the standard, traditional deep packet inspection is losing its edge. Security systems that once relied on reading data payloads are now struggling to keep up with threats they can no longer see because encryption masks the data payload, conventional security methods are now often blind to malicious activity. This paper introduces a machine learning framework designed to detect and classify network attacks within encrypted wire-less traffic using the AWID dataset. By focusing on flow-based and statistical features rather than the hidden content itself, our approach maintains high-level security without compromising user privacy and evaluated three distinct models using MATLAB: a Linear Support Vec-tor Machine (SVM), a Coarse Decision Tree, and a tri-layer Artificial Neural Network (ANN). Our findings reveal that while the Coarse Tree struggled with underfitting (41.67% accuracy) and the SVM reached 87.89%, the tri-layer neural network achieved a near-perfect accuracy of 99%. These results, validated through precision, recall, and F1-score metrics, demonstrate that deep learning architectures are uniquely equipped to overcome the challenges of modern encrypted environments and provide robust network protection.**

***Index Terms*— Encrypted Traffic, Intrusion Detection System (IDS), AWID Dataset, Machine Learning, Neural Networks, SVM, Network Security, Cyber Attacks.**

I. INTRODUCTION

The way people use encrypting protocols like TLS and VPNs has really changed how we think about privacy on the in-ternet. These technologies keep user data safe. They also make it hard for old systems that detect intrusions to work properly. This is because these old systems look at the data to find threats and when the data is encrypted, they cannot do that. Now people who launch cyber-attacks are using this to their advantage hiding in encrypted channels to get past security and launch complicated attacks that older systems cannot detect.

To be able to see what is going on again without hurting people's privacy machine learning is being used. Of trying to decrypt the data machine learning lets us look at how the traffic behaves using patterns to find bad activity. This paper is about a framework that can tell when there is a network attacks in encrypted traffic using something called the AWID dataset. authors tried out a few models like Linear SVM and Decision Trees but mostly focused on making a special kind of neural network with three layers. Our research work demonstrates that this model is highly effective in identifying the patterns in encrypted data that traditional security tools always miss.

Grenze ID: 01.GIJET.12.2.674_26

We are talking about encryption protocols, like TLS and VPNs. How they affect digital privacy and machine learning is helping us with that.

II. Literature Survey

Recent advancements in encrypted traffic analysis have been shifted away from decryption, focusing instead on the pow-er of machine learning (ML) and deep learning (DL) to identify threats through pattern recognition. Much of the current literature explores specialized detection environments; for instance, Ye et al [1] focused on securing WLAN positioning systems via Wi-Fi hotspot tags, while Han et al. (2025) investigated the unique challenges of malicious traffic within 5G networks. The core point of this research often centers on architectural innovation. Researchers like L. Wang et al [2] and Liu et al [3] have moved toward graph-based models—using Graph Convolution Attention Networks and GNNs—to achieve finer classification, while others like Bahlali et al [4] and Meng et al [5] have utilized autoencoders and Varia-tional Autoencoders (VAEs) to spot the subtle anomalies of zero-day attacks.

Optimization and efficiency also remain high priorities in the field. Z. Wang et al [6] leveraged the Information Bottle-neck method to refine CNN performance, and Li et al [7] introduced meta-learning to handle "few-shot" scenarios where attack data is scarce. Beyond pure accuracy, the conversation has recently expanded to include the reliability of these systems. Maung et al [8] and Aghabagherloo et al [9] have addressed the growing threat of adversarial attacks, proposing robust feature extraction and encrypted embeddings as a defense. Furthermore, Senevirathna et al [10] emphasized that for these intelligent systems to be truly integrated, they must be explainable (XAI) to ensure human trust and accountabil-ity. Together, these works—ranging from blockchain-based encryption (Zhang, 2025) to secure UAV authentication (Yoon, 2025)—form a diverse landscape that proves encrypted traffic can be managed safely without exposing private payload data.

III. Methodology

As Shown in Fig.1, The workflow presents a structured pipeline designed to transform raw network traces into actionable security insights. The following sections detail the process from data acquisition to the final application development.

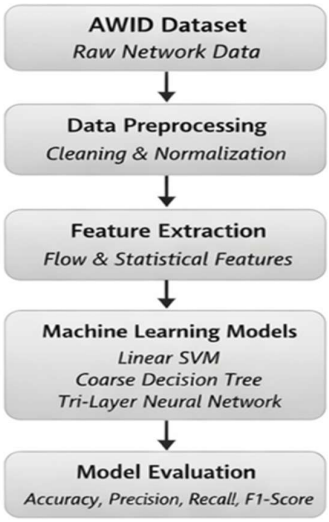

Fig. 1. Workflow diagram

A. Dataset Acquisition

It started with the Aegean Wi-Fi Intrusion Dataset, which we got from Kaggle. The Aegean Wi-Fi Intrusion Dataset was the basis for our training and evaluation and version called AWID-CLS. This version of the Aegean Wi-Fi Intrusion Da-taset helps with classifying things into groups. When traffic is encrypted, you cannot see what is inside. So, authors looked at things, like packet information and flow details to find bad behaviour and did this without messing with the encryption rules of the Aegean Wi-Fi Intrusion Dataset.

B. Data Preprocessing

The raw network data we got was not very clean. It had a lot of unnecessary items and some illogical values. This can make the model not work well. So, we cleaned up the data by eliminating unnecessary information and fixing the values and also changed the labels into numbers so the computer could understand them. Then we made sure all the attributes were on the level so they could be compared easily. To make sure the data was used correctly to split it into two groups: one for training and one, for testing before we made all the values standard.

$$X_{norm} = \frac{X-\mu}{\sigma} \quad (1)$$

where μ and σ represent the mean and standard deviation, respectively.

C. Model Training

Authors wanted to see how well some algorithms worked so they tried out a few using the MATLAB Classification Learner App. This app helped us compare a Linear Support Vector Machine, a Coarse Decision Tree and a Tri-layer Artificial Neural Network. We did this so we could see which model worked best with encrypted traffic and tested the Linear Support Vector Machine, the Coarse Decision Tree and the tri-layer Artificial Neural Network at the same time so it was a fair test. This was a step because it helped us figure out which mathematical structures were good at dealing with the complexities of encrypted traffic, like the Artificial Neural Network and the Linear Support Vector Machine.

D. Feature Selection & Optimization

Our goal is to improve system performance and utilize details. So, we used some methods like looking at how things are connected finding the main parts of the data and picking what is important one by one. When we found the important details we made our system better, by trying different settings and testing it many times with different groups of data. This helps our system work well with data it has not seen before. We conducted this to ensure our data system can manage unfamiliar data and that it remains dependable.

F. MATLAB-Based Application Development

To get the research out of the theory stage and into use we made a working interface using MATLAB App Designer. This thing we made lets people put in Wi-Fi traffic data and get predictions about the network right away. The application shows if everything is "Normal" or if there is an "Attack" happening. It also shows how sure it is, about this and some pictures to help understand the data. This makes the system easy to use in the world.

IV. DATASET

For this research we picked the Aegean Wi-Fi Intrusion Dataset, which's a well-known benchmark. The Dataset is good at showing what normal and bad wireless traffic looks like and used the AWID-CLS part of the Aegean Wi-Fi Intrusion Dataset. This let us see if our system can tell the differences between types of attacks like injection, flooding and impersonation. The Aegean Wi-Fi Intrusion Dataset is a test because it is hard to work with. The traffic is encrypted so we cannot look at the content. We had to look at things like how big the packets are, how strong the signal is and when the packets were sent to find threats.

The Aegean Wi-Fi Intrusion Dataset is also tricky because it has a lot of information and some types of attacks are more common than others. We chose the Aegean Wi-Fi Intrusion Dataset because it is, like world networks, which can be messy and unpredictable.

A. Linear Support Vector Machine (SVM)

Authors used a Linear Support Vector Machine (SVM) as our starting point. Its primary job was to find a clear boundary that could separate the different traffic classes, giving us a standard to measure our more complex models against. It does this by taking the input data and putting it into a space with dimensions. Then it finds the line that leaves the space between the closest data points of each class. The Linear Support Vector Machine makes decisions based on the following formula:

$$f(x) = sign(w \cdot x + b) \quad (2)$$

where w is the weight and b is the bias. When we train the model, it tries to solve a problem that makes the weight as small as possible. This helps the model work well with data not just the data it was trained on. In life the Linear SVM is very good at handling data with many dimensions, like AWID. But it has a problem: it can

only draw straight lines. Be-cause of this it often has trouble understanding the relationships in encrypted network attacks. This is not good because these attacks are often very complicated and do not follow lines. This is why we needed to use an advanced approach like a neural network. The Linear Support Vector Machine just could not handle the relationships, in the data.

B. Coarse Decision Tree

The Coarse Decision Tree. This is a rule-based model that sorts data by finding decision points. It takes all the data. Uses things like the Gini index or entropy to figure out the best way to divide the data into groups. This keeps going until the tree gets to a point like it gets too deep or there are not enough samples. We call this model "coarse" because we made it simple on purpose. It doesn't have a lot of layers. Splits. This makes the model really fast and easy to understand. This simplicity was a big problem in our research. The model was not good at finding the patterns, in encrypted network at-tacks. This was because it was not complex enough. The Coarse Decision Tree had trouble with this. As a result, it did not work well. It was actually our performing model. The Coarse Decision Tree was just not that good enough to handle the complicated encrypted traffic we see today.

C. Tri-Layer Neural Network

The authors made a Tri-Layer Neural Network Fig.2, in to catch the relationships that regular models usually do not see. This network is special because it can look at things in a way that is not limited by lines. It can find the patterns in en-crypted traffic that are hard to see. The network has a main part, a part where we put the information in two hidden parts and a part where we get the results. We use information from the AWID dataset. Put it into the input part. Each little piece of the input part looks at a thing about the traffic. Then this information goes through the parts. These hidden parts use something called the Rectified Linear Unit to help the network understand the information better. The way we set up the network helps in figuring out which things are the signs of a problem with the network. It takes the information and turns it into a powerful tool to find threats. The Tri-Layer Neural Network is really good, at looking at the results to see what is going on and it helps us make sense of the traffic patterns.

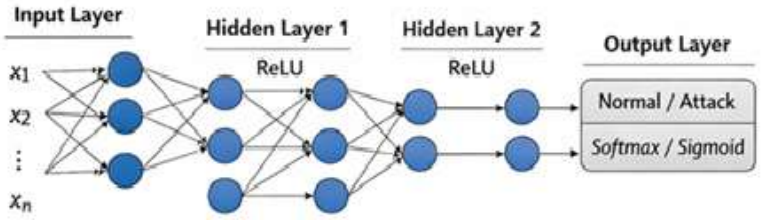

Fig. 2: Tri-Layer Neural Network Architecture

The input layer gets the information from the AWID dataset. The information contains a list of features. Each part of the input layer looks at one feature from the list. The hidden layers then take this information change it in the ways that are not straight forward. They use helpers like the Rectified Linear Unit, which is also called ReLU for short. This helps the network learn to understand the features in a complicated way. The output layer then takes all the changed information. Makes a final decision. It usually uses a math function called sigmoid when it has to choose between two things. It uses a SoftMax function when it has to choose between many things. The AWID dataset is used for training the network and the output layer produces the result based on the information, from the AWID dataset.

The mathematical flow of the forward propagation process is defined as follows:

$$Z_1 = W_1 X + b_1, A_1 = \text{ReLU}(Z_1) \quad (3)$$

$$Z_2 = W_2 A_1 + b_2, A_2 = \text{ReLU}(Z_2) \quad (4)$$

$$Z_3 = W_3 A_2 + b_3 \quad (5)$$

$$\hat{Y} = \sigma(Z_3) \quad (6)$$

The weights and biases of each layer are shown by W and b. The sigmoid activation function is used for binary classification, which is shown by σ. The output layer gives the result, which says if the network traffic is normal or malicious. When the model is being trained it changes its weights using the backpropagation algorithm. This algorithm calculates the gradient of the loss function for each parameter. The loss is usually calculated using the binary cross entropy function. The model employs optimization methods such as descent or Adam optimizer to reduce this loss. As the model improves, it enhances its ability to classify network traffic. The Tri-Layer Neural

Network is particularly suitable for this research because it can capture non-linear feature interactions and learn intricate representations. This leads to improved detection capabilities in comparison to conventional ML models. The Tri-Layer Neural Network excels at classifying encrypted traffic due to its ability to identify non-linear patterns and intricate feature interactions. It differs from other machine learning models as it is capable of learning intricate representations of network behaviour. This provides us with precise outcomes and need to modify certain settings and apply techniques such as validating the model with varied data to pre-vent it from being overly specific and to guarantee its effectiveness with fresh data. The Tri-Layer Neural Network excels in learning and classifying network traffic due to its characteristics.

V. Results and Discussion

Authors made a system to detect people trying to get into a network. We used a set of data called AWID-CLS and a tool called MATLAB to build it and tried out a few models to see how well they worked. These models were a Linear Support Vector Machine, a Decision Tree and our own special Tri-Layer Neural Network. We wanted to see if the system could handle kinds of threats like Botnet, Deauthentication, Evil Twin, Rogue Access Point and SQL Injection. What we found out was that some ways of using machine learning were better than others. The Linear Support Vector Machine worked well. It got 87.89% of the answers right. This means it is good when there are a lot of things to look at. It had trouble with things that were hard to tell apart and it made mistakes with "Evil Twin" and "Rogue Access Point" attacks. The simple Decision Tree did not work well only getting 41.67% of the answers right. This happened because it was too simple and could not make decisions about the secret traffic data. On the hand the Tri-Layer Neural Network was almost perfect get-ting 99% of the answers right. It worked better than the models. It was able to understand the traffic and find the bad activities with a high degree of accuracy across different kinds of network attack detection system threats. We made a MATLAB tool that can predict what will happen over time as shown in Fig.3. This tool lets users put in Wi-Fi traffic data and get updates right away along with scores that show how confident the tool is. This illustrates that highly accurate models can evolve into practical, user-friendly tools for identifying intrusions within a network attack detection system.

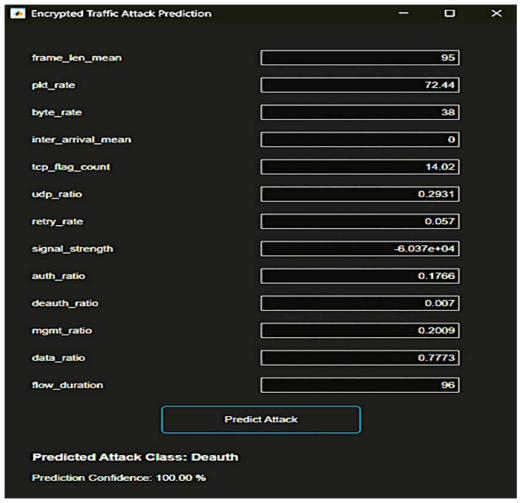

Fig.3. MATLAB App Environment

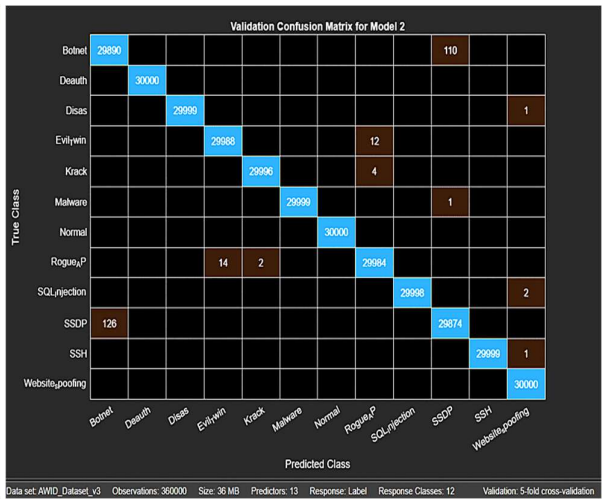

Fig.4. Confusion Matrix of the proposed work

In Fig.4. The validation results of the Tri-Layer Neural Network model are really good. This model can tell the difference between the 12 classes of the AWID dataset. Most of the time it gets it right. For example, it can tell what is Deauthentica-tion and what's Normal traffic. It can also tell what is Website Spoofing and what's SSH. The model gets it right every time for these classes. There are not mistakes. Sometimes the model gets confused between things that're similar like Bot-net and SSDP or Evil Twin and Rogue Access Point. This is not surprising because these things can look a lot alike. The model is very good at figuring out what is what. It looks at a lot of samples 360,000 of them. It does not make many mistakes. This means the Tri-Layer Neural Network model is very good, at detecting and classifying encrypted network traffic. The Tri-Layer Neural Network model can do this because it is strong and it can generalize well. The Tri-Layer Neural Network model is also robust.

A. Comparative Analysis

In Table.1, When we look at ways to categorize encrypted traffic, we see that old methods and simple learning models do a job, but not a great job. For example, Ye and his group came up with a way to find spoofing in networks, Shan and Ma made a model to detect intrusions using data and Li and his team tried a new way to classify encrypted traffic with just a few samples. These methods usually get it right 80 to 93 percent of the time but they have problems because they do not have enough data to train with and they do not have enough features. On the hand the new Tri-Layer Neural Network model does a lot better getting it right 99 percent of the time which shows that it is really good at finding complex pat-terns in encrypted traffic data. The Tri-Layer Neural Network framework is really good at categorizing encrypted traffic. This is because it can find patterns in encrypted traffic data, which makes the Tri-Layer Neural Network model work well. The Tri-Layer Neural Network model is very good, at what it does which is categorizing encrypted traffic.

TABLE I: COMPARISON OF ACCURACY WITH EXISTING WORK

Author	Method Used	Accuracy / Performance
Ye et al. (2025) [1]	Spoofing Attack Detection in WLAN Positioning Systems	93%
Shan & Ma (2025) [13]	Big Data-Based Intrusion Detection Model	92%
Li et al. (2025) [19]	Few-Shot Learning (Meta-Learning Approach)	90%
Proposed Method	Tri-Layered Neural Network	99%

VI. CONCLUSION

The proposed work shows a best way to find and classify network attacks in encrypted traffic. The authors used the AWID-CLS dataset and MATLAB to accomplish this and found out that by looking at the features of the traffic we can find behaviour without having to decrypt the data. Compared to machine learning methods with new deep learning methods. The old methods, like Linear SVMs and Coarse Decision Trees are okay. Only the Tri-Layer Neural Network was good enough to find the complex patterns in encrypted threats with a precision of 99.99%. What is also important is that our model is efficient and can be used in situations. authors made sure to clean up the data and pick the features, which helps our model work well. MATLAB application that can be used in life, which is a big step forward. This appli-cation can show what is happening in real time. With increasing amounts of network traffic being encrypted, our research suggests that creating techniques to identify harmful actions is a successful strategy and will be vital for ensuring safety and security in our online space.

FUTURE WORK

The suggested Tri-Layer Neural Network excels at delivering accurate results. Further steps can be implemented to im-prove the Tri-Layer Neural Network's capability and effectiveness in the community. To ensure the Tri-Layer Neural Net-work model performs effectively, we must evaluate it using larger and more diverse sets of real-world data that represent various traffic scenarios. We believe it would be beneficial to employ additional sophisticated methods such as Convolu-tional Neural Networks for examining spatial patterns, Recurrent Neural Networks for analyzing temporal events, and Graph Neural Networks for exploring connections between entities. This can help us get an understanding of the features we are looking at with the Tri-Layer Neural Network. One big problem we still need to solve with the Tri-Layer Neural Network is that the data we are working with is not balanced. This means we need to find ways to make the Tri-Layer Neural Network better at finding types of attacks with the Tri-Layer Neural Network.

We can try adding data or making samples to help with this for the Tri-Layer Neural Network and also want the model work fast and in time. For this, it is necessary to ensure the model integrates effectively with the existing systems used for network monitoring. It is essential to ensure that individuals comprehend how the Tri-Layer Neural Network formulates its decisions. This is where Explainable AI comes in. By making the Tri-Layer Neural Network models decision-making process clear we can make the Tri-Layer Neural Network more trustworthy and easier to use especially when it comes to cybersecurity infrastructure.

REFERENCES

[1] Ye et al., “Detection of Spoofing Attacks in WLAN-Based Positioning Systems Using WiFi Hotspot Tags,” IEEE Transactions on Mobile Computing, pp. 1–14, 2025.

[2] L. Wang et al., “ACG: Attack Classification on Encrypted Network Traffic Using Graph Convolution Attention Networks,” IEEE Transactions on Information Forensics and Security, vol. 19, pp. 1–13, 2025.

[3] P. M. Maung et al., “Hindering Adversarial Attacks with Multiple Encrypted Patch Embeddings,” IEEE Transactions on Image Processing, vol. 34, pp. 1–12, 2025.
[4] Y. Zeng et al., “Monitoring and Analysis of Encrypted Attack Traffic Based on Machine Learning,” IEEE Access, vol. 13, pp. 1–10, 2025.
[5] R. Bahlali et al., “Malicious Encrypted Network Traffic Detection Using Deep Auto-Encoder with a Custom Reconstruction Loss,” IEEE Transactions on Network Science and Engineering, vol. 12, pp. 1–14, 2025.
[6] Z. Li et al., “Known-Plaintext Attacks to Thumbnail-Preservation Encryption Using Pix2pix Generative Adversarial Network,” IEEE Transactions on Information Forensics and Security, vol. 20, pp. 1–12, 2025.
[7] X. Zhang et al., “SBiPMU-Based Power Data Encryption Traffic Attack Forecasting,” IEEE Transactions on Smart Grid, vol. 16, pp. 1–9, 2025.
[8] R. Zhang, “Research on the Encryption Model of Network Security Information Transmission Based on Blockchain,” IEEE Access, vol. 13, pp. 1–14, 2025.
[9] K. Yoon et al., “Security Authentication System Using Encrypted Channel on UAV Network,” IEEE Transactions on Vehicular Technology, vol. 74, pp. 1–10, 2025.
[10] Gujarathi et al., “Advanced Encryption Using Generative Adversarial Network for Enhancing Security of Non-Fungible Tokens (NFTs),” IEEE Access, vol. 13, pp. 1–15, 2025.
[11] H. Weerasena et al., “Lightweight Encryption Using Chaffing and Winnowing with All-or-Nothing Transform for Network-on-Chip Architectures,” IEEE Transactions on Computers, vol. 74, pp. 1–12, 2025.
[12] Q. Meng et al., “Detection of Unknown Attacks Through Encrypted Traffic: A Gaussian Prototype-Aided Variational Autoencoder Framework,” IEEE Transactions on Information Forensics and Security, vol. 20, pp. 1–15, 2025.
[13] J. Shan and H. Ma, “Optimization of Network Intrusion Detection Model Based on Big Data Analysis,” IEEE Access, vol. 13, pp. 1–15, 2025.
[14] T. Senevirathna et al., “Enhancing Accountability, Resilience, and Privacy of Intelligent Networks With Explainable AI,” IEEE Transactions on Network and Service Management, vol. 22, no. 1, pp. 1–20, 2025.
[15] Aghabagherloo et al., “Unveiling Illusionary Robust Features: A Novel Approach for Adversarial Defenses in Deep Neural Networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 47, pp. 1–14, 2025.
[16] Z. Wang et al., “Robust Classification of Encrypted Network Services Using Convolutional Neural Networks Optimized by Information Bottleneck Method,” IEEE Transactions on Network Science and Engineering, vol. 12, pp. 1–10, 2025.
[17] S. Alhazbi et al., “LLMs Have Rhythm: Fingerprinting Large Language Models Using Inter-Token Times and Network Traffic Analysis,” IEEE Open Journal of Communications Society, vol. 6, pp. 5050–5065, 2025, doi:10.1109/OJCOMS.2025.3577016.
[18] G. Han et al., “AI-Based Malicious Encrypted Traffic Detection in 5G Data Collection and Secure Sharing,” Electronics, vol. 14, no. 51, pp. 1–24, 2025.
[19] Z. Li et al., “Unlocking Few-Shot Encrypted Traffic Classification: A Contrastive-Driven Meta-Learning Approach,” Electronics, vol. 14, no. 4245, pp. 1–16, 2025.
[20] Z. Liu et al., “Fine-Grained Encrypted Traffic Classification Using Dual Embedding and Graph Neural Networks,” Electronics, vol. 14, no. 778, pp. 1–19, 2025.