# Deep Learning-Based Models for Malicious File Segregation in Naval Networks

Aarzoo Gosain[1], B.E. (I.T.), Aseem Gosain[2]*, Gold medallist, B.Tech (Electrical), M.Tech (Power Electronics), Commander Rakesh Kumar Gosain, Retd.[3], B.Tech. (Electrical), M.Sc. (Physics)

[1] Netaji Subhas Institute of Technology (University of Delhi), New Delhi, India

[2] Indian Institute of Technology, Varanasi, India

[3] Veteran, Indian Navy

* Corresponding Author. Email: aseemgosain@gmail.com

**Synopsis**

**Background**: The naval industry faces escalating cyber threats from sophisticated malware attacks such as denial-of-service, spyware, and ransomware, which jeopardise confidential information and operational security. Traditional security measures often struggle to accurately detect encrypted or obfuscated malicious files within network traffic, especially in real-time scenarios. The legacy entropy-based file segregation model (Gosain & Gosain, 2022) deployed aboard combatants detects encrypted malware but suffers from high false positive rates and manual throughput limits. A fully automated Malware Arresting and Recommendation System is proposed to address these challenges. This model leverages deep learning-based models, such as CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory) Network, BERT (Bidirectional Encoder Representations from Transformers) and its variants, fine-tuned to analyse network file traffic. By treating files as sequences akin to natural language, the model employs advanced natural language processing techniques to extract semantic embeddings, enabling the effective segregation of suspected malicious files from benign ones. The proposed model uses deep neural networks to learn byte-level and sequential patterns of benign and malicious Portable-Executable (PE) files.

**Methodology**: The proposed model utilises deep learning-based models to generate high-dimensional embeddings that capture the intricate patterns and dependencies within file sequences. Training is conducted on comprehensive datasets comprising malicious and benign files, including encrypted and polymorphic malware samples. Four candidate models, namely, CNN, three-layer LSTM, BERT-Large and CodeBERT, are trained and evaluated on the 11,000-file corpus with a ratio of 1000 benign: 10,000 malware files (DikeDataset 2022). Class imbalance is mitigated through random subsampling of the majority class.

**Results and Observations**: The proposed model, built upon deep networks, achieves a much higher detection accuracy than the traditional entropy-based segregation model, all the while reducing the false positive rates. The model built upon the CodeBERT framework attained the highest balanced accuracy of 95.4 %. The least inference time of 0.1 milliseconds per sample was observed in the CNN-based model, outperforming the earlier entropy-based model and delivering 20 times lower compute load than the transformer baselines.

**Conclusion and Applications**: In conclusion, integrating a hybrid CNN-CodeBERT Recommendation model into network traffic analysis represents a significant advancement in naval cybersecurity defence mechanisms. With a fast preliminary screening by the CNN head and a deep inspection of the flagged files by the CodeBERT body, this hybrid model will be able to accurately and efficiently detect the malicious files within the encrypted and regular network traffic, enhancing the security of command and control communications. The model's real-time capabilities and adaptability make it suitable for various applications, including secure communications in weapon systems, unmanned vehicle coordination, hypersonic glide vehicles, and missile guidance systems. Operation in denied, degraded or disrupted SATCOM scenarios is preserved because all inference executes locally without cloud offloading. By leveraging advanced natural language processing and deep learning techniques, this research contributes to strengthening cyber defence measures within naval operations and can be extended to other critical infrastructure sectors requiring robust security solutions.

*Keywords:* Transformer Language Models, BERT, CodeBERT, Natural Language Processing, Cyber Defence, Naval Networks, Secure Communication, Network Security, Real-Time Analysis.

---

**Authors' Biography**

**Aarzoo Gosain** is an alumnus (batch of 2021) of Netaji Subhas Institute of Technology (University of Delhi). She holds a Bachelor of Engineering in Information Technology and has published a conference paper in the IET journal regarding AI-based real-time noise suppression. She has led several projects in the fields of Artificial Intelligence and Cybersecurity during her course of undergraduate study.

**Aseem Gosain** is a Gold Medallist alumnus of the Indian Institute of Technology (BHU), Varanasi, Class of 2018 (Electrical engineering and Power electronics). He is currently working as a Consulting Sr. Data Scientist and Applied AI Engineer with a focus on recommender systems, information extraction and sequence modelling.

**Cdr. Rakesh Kumar Gosain, Ret.** is an Indian Navy veteran and has served as a Scientist 'F' in PMO, India. He is a marine electrical officer having versatile professional experience ranging from Naval electrical and weapon systems to project and contract management of multi-billion-dollar G to G projects of strategic interest to his credit. His area of expertise is missiles and associated Close-in Weapon Systems (CIWS) and Sensors.

## 1. Introduction: An advancement to the manually-operated file segregation model

Any computer network deals with a humongous amount of incoming and outgoing network file traffic, which may or may not be safe for the network. Therefore, it becomes a priority to prevent such malicious files from entering the network. A malware attack, which is executed by sending a malicious file to the intended computer network, either hides its presence in or masks itself as genuine data. Malicious software can use encoding or encryption techniques for imitation purposes.

After deciding upon the method for sending the malware, the prepared malicious file is then floated onto the incoming file traffic of the computer system of the intended network. Malware attacks may be carried out in many ways. With Spyware attacks, the case is even worse because the malware hides its presence and works quietly in the background, so its presence is not discovered until it is too late. The onus lies on the Packet Filtering mechanism of the firewall to prevent such malicious files from entering the network. Also, the Deep Packet Inspection (DPI) method of packet filtering is sometimes infeasible, as opening and checking each file in a secure environment exhausts many resources of the computer system.

To reduce the burden on the firewall and to prevent malware attacks, a manual file segregation model was introduced in the previous research, which used the entropy concept to measure the amount of information that is present within the data characters in a file (Shannon 1948). However, there were major drawbacks to this file segregation model. These were:

1. The file segregation model was to be operated manually.
2. The pipeline analysed only one statistic per file, i.e., Shannon's entropy value of the given file.
3. The model struggled with borderline encoded files, mainly PDF files having 6-7 bits/byte entropy value.
4. The model's balanced accuracy (a metric for obtaining accuracy reliably on the imbalanced dataset) was very low, only 51.53%.

In this research, a fully automated Deep Learning-based Malware Arresting Recommendation System is proposed, which is an Artificial Intelligence-based upgrade to the manually-operated file segregation model. This Recommendation model will prove to be beneficial in strengthening the existing firewall mechanisms of the Naval Command and Control Secure Systems, as it is capable of identifying malicious files in a much shorter period, with minimal latency and greater accuracy.

## 2. Related Work

Table 1: Research Literature on Deep Learning-based Malware File Classification

| S.No. | Research Reference | Approach | Dataset used | Limitations for Naval use |
|---|---|---|---|---|
| 1. | Wang et al. 2017 | CNN scans raw bytes and learns patterns | USTC-TFC2016 | Very accurate, but needs a GPU and a full packet stream |
| 2. | Hwang et al. 2019 | LSTM reads packet features and spots bot traffic | USTC-TFC2016 | Very good on time-series data, but testing is conducted on limited data |
| 3. | Gosain & Gosain 2022 | Entropy check flags files whose randomness is high | Netresec | Runs fast but misses low-entropy malicious files, needs a manual review |
| 4. | Cao, Luo and Wu 2022 | GAN + CNN converts PCAP files into images and trains the CNN | Malware Capture Facility project | Accurate, but image conversion and GAN training take up a lot of computational time |
| 5. | Ullah et al. 2024 | Large transformer scans every byte with self-attention | CIC-IDS2017, UNSW-NB15, NSL-KDD | Huge model (250M parameters) - too heavy for ship CPUs |

It can be seen that in the first research, a small CNN is used to scan the file and identify the malware (Wang et al. 2017). Although it scores a very high accuracy, it also needs a GPU (Graphics Processing Unit), which the Naval ships and submarines do not carry.

The main advantage of the recommendation model proposed in this research is that it is a hybrid model. It makes use of a two-layer CNN head that scans file bytes sequentially, learning the patterns of both benign and malicious files which are present in the training data. Afterwards, these flagged files are made to pass through the body of the model, which consists of the CodeBERT framework. Thus, malware-ridden files are segregated and prevented from entering the network.

The proposed recommendation model delivers high balanced accuracy and significantly lower latency. It needs no GPU or internet connection, so it fits easily on ships and submarines, even when SATCOM is denied.

## 3. Methodology

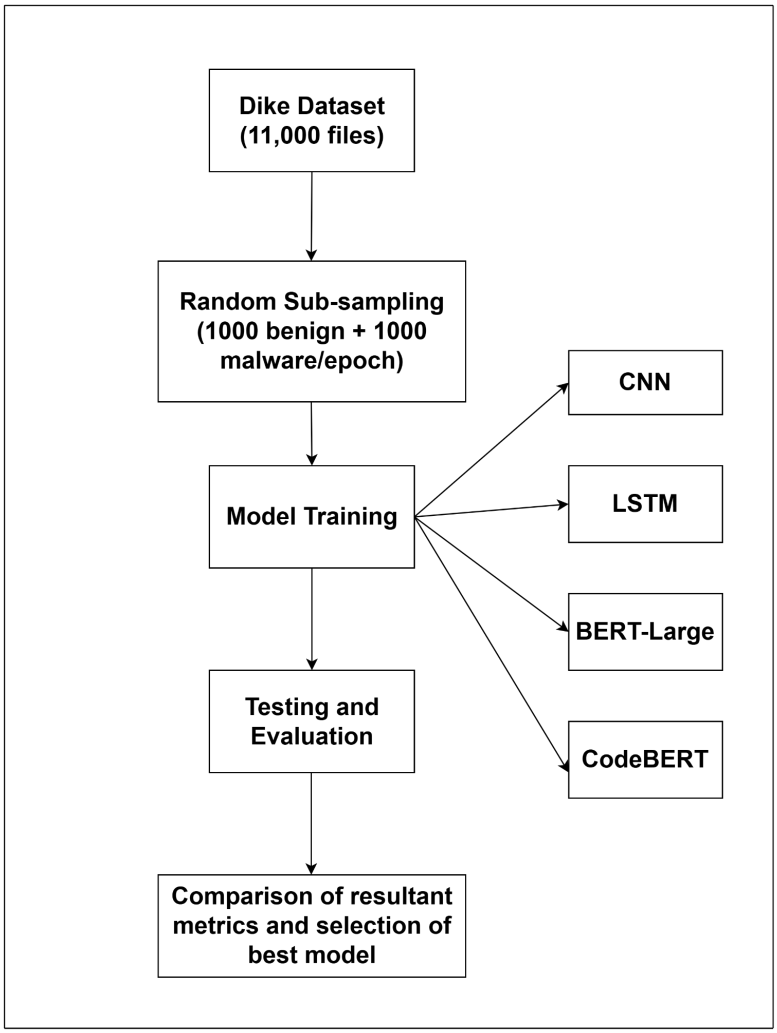


Figure 1: Process of Training and Testing Deep Neural Networks for Malware File Classification

The procedure used for training and testing of the deep neural networks for malware file classification is represented in Figure 1:

1. **Collection of Data:** The first step for any neural network model training is the collection of data. For this research, the Dike Dataset has been used. This dataset consists of 10,000 known malware files (ransomware, spyware, viruses, trojan horses, etc.) and 1000 legitimate service files.
2. **Random sub-sampling for balancing the classes:** As the ratio of benign to malicious files is 1:10, it becomes imperative to balance the number of files in both these classes. For this purpose, the process of sub-sampling is used. The 50:50 mix stops the model from leaning toward the majority class. Afterwards,

the training and test data are split, with 85 % of the data being reserved for training and 15 % for final testing to measure how well each model copes with unseen traffic.

3. **Model training:** Feed balanced mini-batches through CNN, LSTM, BERT-Large and CodeBERT architectures, letting each learn byte-level patterns that distinguish benign from malicious files.
4. **Testing and evaluation of the models:** Run the fixed test set once per model, logging accuracy, balanced accuracy, precision, recall, F1-score and inference time. These terms are explained below.
    a. **Accuracy:** How often is the model right overall? If it checks 100 files and gets 97 of them correct, its accuracy is 97 %.
    b. **Balanced accuracy:** The average of 'how well it finds malicious files' and 'how well it lets benign files through'. This methodology keeps the score fair when one class (malware files in this case) is much bigger than the other.
    c. **Precision:** When the model flags a file as 'malicious', precision tells how often that flagging is true. High precision means few false alarms.
    d. **Recall/Sensitivity:** Of all the real malware hiding in the traffic, recall is the share that the model actually catches. High recall means few misses.
    e. **F1-score:** A number that balances precision and recall. It is the 'sweet-spot' score, which is high only when the model both avoids false alarms and misses very little malware.
    f. **Inference time:** The time the model needs to judge a single file. Short inference time (e.g., 0.1 milliseconds) means it can keep up with fast network links without slowing the traffic.
5. **Comparison of results and selection of the best model:** The models are ranked based on their accuracy and latency.

## 4. Experimental Set-up

### *4.1. Dataset selection*

The dataset used for this research is the Dike Dataset, which is a labelled dataset of 11,000 PE (Portable Executive) and OLE (Object Linking and Embedding) files. The malware-to-benign demarcation in the dataset is 10,000 malware files to 1000 benign files. The malware files consist mainly of ransomware, remote-access Trojans (RATs) and small droppers that plant other payloads.

### *4.2. Data Pre-processing*

Pre-processing of data is to be done to balance the malware and benign classes in the training data, because feeding imbalanced data to the deep neural network model gives biased results leaning towards the majority class. As the malicious files outnumber safe ones by 10:1, we balance the training data with random subsampling. 1000 malware samples are drawn so that their count matches the 1000 benign files. Training then runs on this 50:50 batch. The draw is repeated for the next epoch, so the model gradually sees almost every malware sample but never learns a bias toward the larger class. Treating class imbalance keeps learning fair, avoids the need for synthetic data, and keeps memory use low. For training purposes, every file is sliced into easy-to-handle chunks of 512 bytes.

### *4.3. Neural Network Architecture*

For the purpose of selecting the best deep neural network framework as the backbone of the proposed Recommendation model, four networks, namely, CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory) Network, BERT-Large (Bidirectional Encoder Representations from Transformers) and CodeBERT (pre-trained BERT model to analyse code sequences) were trained and tested on Dike dataset containing labelled benign and malicious files.

1. **CNN:** Uses small sliding filters to spot suspicious byte shapes quickly. It is both fast and light, making it perfect for real-time gateway checks.
2. **LSTM:** Reads the file one byte at a time and remembers what came before, so it can catch patterns that are far apart.
3. **BERT-Large:** A big transformer that looks at every byte in the window at once and learns rich relationships, but needs lots of memory and time.
4. **CodeBERT:** A smaller transformer already trained on computer code. It can be fine-tuned on the malware bytes so that it can start with useful knowledge of how the programs are built.

*4.3.1. Baseline CNN Architecture*

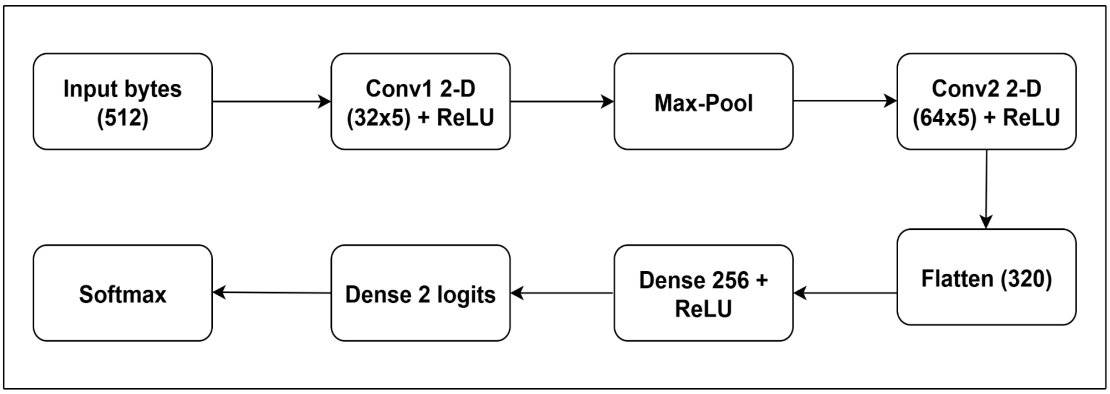


Figure 2: CNN Architecture used in Recommendation Model

A 2-layer CNN is trained on 1.2 million file parameters, having 32 scanners in the first convolutional layer and 64 in the second. The CNN takes in the first 512 bytes of the file as its input. These bytes are like 'raw letters' which are read by the neural network (Cao, Luo and Wu 2022). A set of 32 tiny scanners in the first layer looks at 5-byte windows and flags any small suspicious shapes it sees. The set of 64 scanners in the second layer identifies the greater and more detailed patterns built from the clues generated by the first layer. The 2-D pattern thus obtained is mapped into one long row of 320 numbers so that the Dense layer can read it (Kalash et al. 2018). Dense 2 logits produce two final scores—one for 'malicious', and another for 'benign' nature.

*4.3.2. Three-Layer LSTM Architecture*

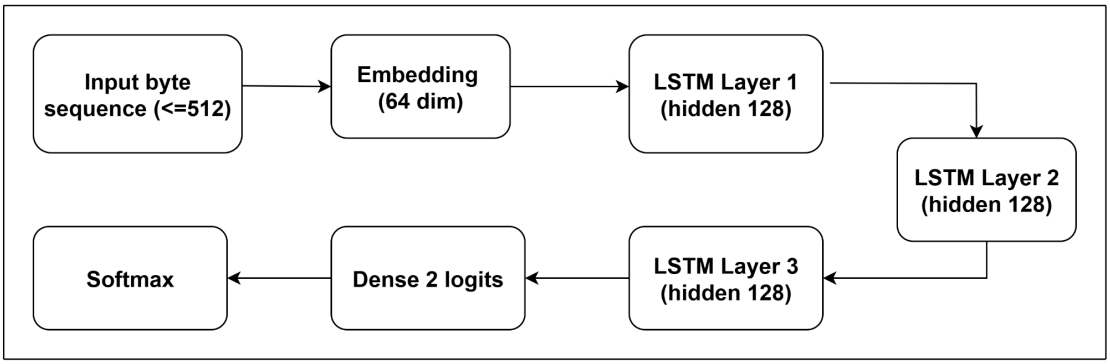


Figure 3: LSTM Architecture used in Recommendation Model

LSTM reads the file one byte at a time, remembering the bytes that came before. The internal memory lets it identify anomalies occurring even at the end of the file sequence, such as a malicious code hidden far from the header. The three stacked LSTM layers help the later layers reason over bigger patterns built by earlier ones. This sequential modelling is expected to catch malware that the CNN might miss, but the extra memory makes it roughly three times slower. It is trained on 2.3 million file parameters (Hwang et al. 2019).

LSTM Layer 1 starts remembering recent bytes and flags quick local clues. Layer 2 remembers the longer, linking clues that are far apart. Layer 3 builds a full-file memory to see big malicious patterns. Dense 2 logits shrink everything to two raw scores.

*4.3.3. BERT-Large Architecture*

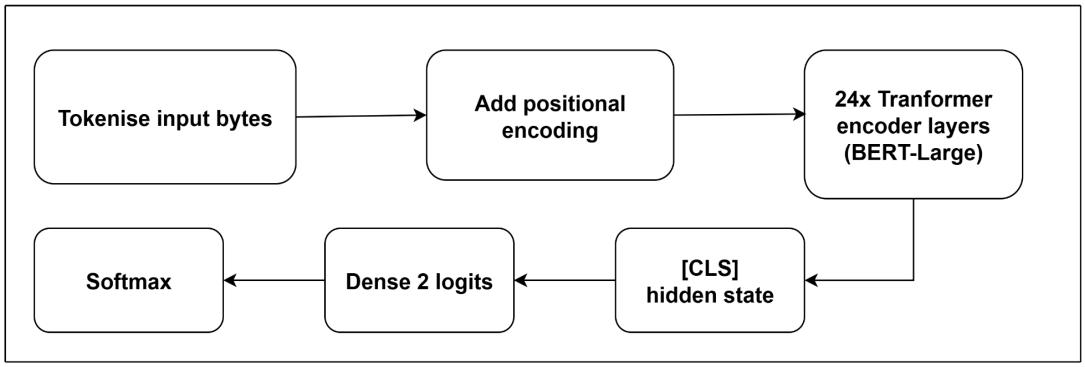


Figure 4: BERT-Large Architecture used in Recommendation Model

BERT model is a pre-trained transformer model that is trained on a large corpus of English language data. It has two sizes, namely, BERT-Base and BERT-Large. BERT-Large has 340 million parameters, and the base model has 110 million (Ullah et al. 2024). In this research, BERT-Large is chosen as one of the candidate frameworks of the Recommendation model. It is fine-tuned for five epochs (5 complete passes) with a learning rate of $2 \times 10^{-5}$ (step size for optimising the training) with early-exit disabled for a fair timing comparison.

BERT looks at all 512 bytes at once and builds a map of which bytes should pay attention to each other. This global view uncovers complex, spread-out hiding tricks. As BERT is pre-trained on 340 M parameters, it needs a lot of RAM, and its latency is 25 milliseconds per file, making it unsuitable for Naval ship CPUs.

*4.3.4. CodeBERT Architecture*

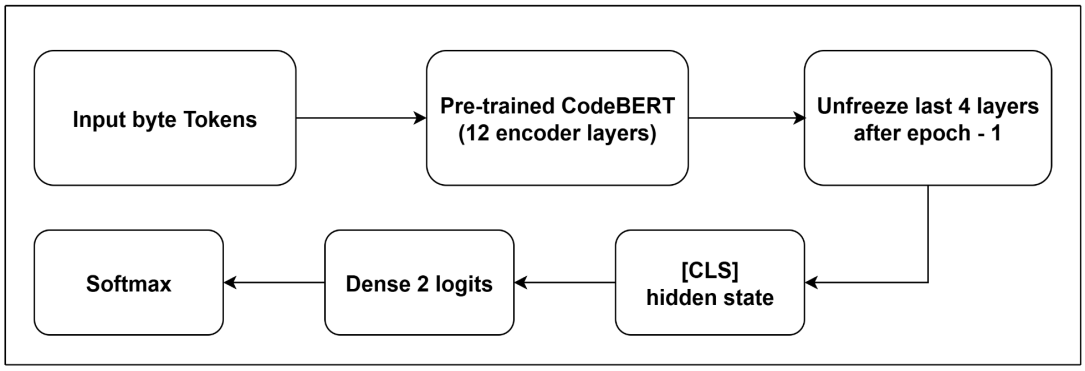


Figure 5: CodeBERT Architecture used in Recommendation Model

The CodeBERT model is a pre-trained transformer model that is trained on both English language and programming language data. It has 125 million parameters. The greatest advantage of the CodeBERT model over the BERT model is that any anomaly that the BERT is not able to interpret because of the anomaly's 'code-like' nature can easily be identified and detected by CodeBERT (Feng et al. 2020).

The input byte tokens function breaks the file bytes into tiny tokens that the model expects (Zhang et al. 2024). In this research, the last four layers are unfreezed gradually (i.e., gradually allowing deeper layers to learn malware features without losing coding skills) after the first epoch itself.

### *4.4. Model Training and Evaluation Details*

The training and evaluation of the deep neural network models have been conducted on an AWS SageMaker instance with a single NVIDIA L4 GPU. AdamW Optimiser is used to adjust each model's gradients during the training so that they improve after each pass. Every model used in this research has an input batch size of 32 files and are trained for five epochs each.

## 5. Results and Discussion

Table 2: Performance of Deep Models on Dike Dataset

| Deep Neural Model | Accuracy (%) | Balanced accuracy (%) | Precision | Recall | F1 Score | False Positive (%) | Inference Time (ms) | Parameter count (Million) |
|---|---|---|---|---|---|---|---|---|
| CNN | 97.8 | 93.5 | 0.988 | 0.987 | 0.988 | 1.1 | 0.1 | 1.2 |
| LSTM | 92.5 | 92.3 | 0.991 | 0.926 | 0.957 | 0.7 | 0.3 | 2.3 |
| BERT-Large | 25.3 | 54.5 | 0.95 | 0.188 | 0.314 | 0.9 | 24.5 | 340 |
| CodeBERT | 97.2 | 95.4 | 0.995 | 0.975 | 0.984 | 0.4 | 13.9 | 125 |

From the results obtained above, it is evident that CodeBERT has the highest balanced accuracy, around 95 % and CNN has the least inference time, i.e., it finishes its check in one-tenth of a millisecond, so that a single core can inspect about ten thousand files per second (fast enough for a busy network). LSTM is accurate but takes three times as long as the CNN. Its built-in memory helps it identify long-range tricks, yet 0.3 ms per file means the same core now tops out around three thousand files per second.

The BERT-Large model is pre-trained on 340 M parameters. Hence, it needs 25 ms per file, which is roughly 250 times slower than the CNN. BERT-Large could not deliver good accuracy because it is naturally pre-trained on English language data, and hence, it is not suitable for this malware file traffic classification research. However, it is important to consider the fact that both the number of training epochs and the dataset were limited, which is not nearly enough to properly fine-tune such a huge model like BERT (Ferrag et al. 2024). Hence, its accuracy is not that great.

In the CodeBERT model, which is originally pre-trained on both English language and programming language data, five epochs and the dataset were sufficient for it to deliver a great balanced accuracy. CodeBERT's balanced accuracy is better than that of the CNN, but the model is much slower in comparison. Most of that delay comes from its 125 M parameters and the resulting larger calculation load for every inference.

The graphical representation of the F1 scores obtained by the deep models is shown in Figure 6 below.

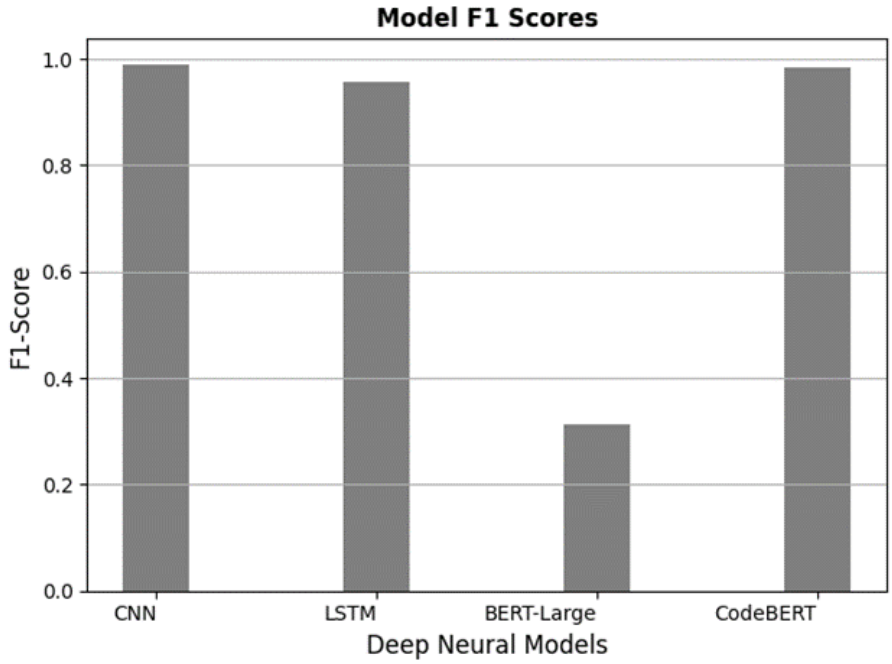


Figure 6: F1 Scores obtained by the Deep Neural Models

The pre-trained large transformer models like BERT and CodeBERT are more suitable for the 'deep-inspection' of malicious file traffic as they can grasp the feature-rich view of the whole data at once, enabling an efficient vulnerability detection as compared to the models which are trained from scratch.

## 6. Proposed Framework of Recommendation Model: Hybrid CNN + CodeBERT

Although CNN is really fast, it can still flag some benign but tricky files as 'malicious', raising false alarms. The CodeBERT model is slower (≈14 ms per file) yet has the highest balanced accuracy. It is much more efficient in the harder cases because it is pre-trained to understand the real program structure (Rahman et al. 2024). The false alarm and delay in traffic situations can be of grave concern if this recommendation model is installed in Naval Command and Control Networks. A fully automated Hybrid CNN + CodeBERT model is proposed in this research. It has a three-stage workflow as described below.

**Stage 1 – Quick look with the CNN:** Every incoming file is first run through the tiny 2-layer CNN. If the CNN is very sure that a file is clean (a very low 'malicious' chance), the file is let go at once. The result is that a large amount of daily traffic is cleared in less than 0.1 ms. No queue builds up on the network's link.

**Stage 2 – Slow, deep-inspection for the 'grey-zone':** Files that the CNN flags as 'malicious' are copied to a side process that runs the CodeBERT framework. Because only a few files reach this step, CodeBERT's 14 ms delay barely dents the overall throughput.

**Stage 3 – Merging the two verdicts:** If CodeBERT marks a file as 'benign', the file is forwarded; otherwise, it is quarantined.

This hybridised model is hoped to cut down false positive rates by 60 % in comparison with the CNN-only model, yet the average end-to-end delay should stay under 0.5 ms (Han, Zeng and Song 2020). The model will be able to answer in a shorter span of time, so that mission data never piles up.

## 7. Naval Applications of the Proposed Recommendation Model

The Denied, Degraded, Disrupted, Intermittent, or Limited (D3IL) conditions are operational environments where communication bandwidth is scarce, compute resources are constrained, connectivity may be unstable, and systems must continue to function under adversarial pressure. These conditions are common in naval operations at sea, especially during combat and/or electronic warfare scenarios. In such settings, traditional cloud-reliant security solutions fail because they require constant high-bandwidth connectivity and abundant processing power.

The proposed hybrid CNN+CodeBERT recommendation model is specifically designed to address these D3IL challenges by being lightweight and entirely local. Packet filtering is performed in two layers: the CNN head conducts a rapid first-pass scan of every incoming packet, using lightweight convolutional filters to dismiss benign traffic with sub-millisecond latency. Suspicious or ambiguous packets are escalated to the CodeBERT body, which provides deep inspection by leveraging its pre-training on code and language structures to identify stealthy or polymorphic malware. This two-tier process ensures that packet-level segregation of malicious from benign traffic occurs locally, without dependence on cloud offloading, thereby maintaining resilience in constrained and contested environments.

By combining the speed of CNN for initial filtering with the deep inspection capability of CodeBERT, the model can ensure that malicious payloads are detected before they impact critical naval functions. The fast/deep inspection capabilities demonstrate the real-world operational utility of the proposed recommendation framework in enhancing the cybersecurity resilience of the Naval Systems. Apart from its obvious use as a network pre-filter, below are three unique cases where this model has further practical impact.

### *7.1. Smart Decoy System*

Modern anti-missile defence systems employ AI-driven swarms of drones and decoys to confuse incoming threats. These systems rely on secure, real-time command and control communications. Malicious files injected into the decoy swarm's coordination channels could disable the swarm or expose the host ship. The filtering of this traffic happens in two stages. Firstly, large volumes of command packets are screened at sub-millisecond speeds, ensuring the decoys are not overwhelmed by false data. Secondly, files that are flagged as suspicious are examined for their code-like structures to detect any hidden malware. At the packet level, this approach ensures that every instruction entering the swarm network is verified before execution, preserving mission effectiveness even when bandwidth is constrained or communications are degraded. This layered defence ensures the swarm continues to operate effectively without interruption or compromise.

### *7.2. Anti-Jamming Technology using the Proposed Model*

Shipborne communication and guidance systems often employ dynamic frequency hopping or adaptive band-pass filtering to counter electronic jamming. Adversaries can attempt to degrade these systems by inserting malicious payloads that mimic or distort legitimate control instructions, confusing the jamming mitigation process. The hybrid recommendation model filters incoming control traffic in real time. The CNN quickly eliminates benign traffic, while suspicious instructions undergo deeper inspection by CodeBERT to confirm whether they contain embedded malware. This methodology ensures that even when communication links are unstable, the system only processes clean, validated packets. By ensuring only safe control packets reach the adaptive filtering algorithms, the model preserves the reliability of anti-jamming measures and prevents adversaries from using malware to destabilise the communication channels.

### *7.3. Dynamic Filtering of Sea and Ground Clutter in Radar*

Radar operations are adversely affected in determining the position and speed of low-altitude projectiles due to weak signal strength arising from sustained sea or ground clutter. Radars have a static threshold set in their receivers. Attackers may inject malicious signal patterns into radar data streams, creating false clutter or masking real threats. The proposed recommendation model, being a sequence analyser in essence, can be fine-tuned on radar data to predict a dynamic threshold (probability of actual clutter) for the radar's data handling pipeline, so that the Signal to Noise Ratio (SNR) is boosted significantly and malware can be filtered out before it interferes with clutter suppression algorithms. This will enhance the operational reliability and efficiency of radar systems in identifying and detecting stealthy low-flying targets. The dynamic modulation of the Radar threshold can operate offline and improve the efficiency of detection by reducing the false-alarm rate in clutter/jamming scenarios and protecting against cyber-augmented deception tactics.

## 8. Conclusion

In the previous research, a manually operated file segregation tool was developed to segregate malicious files from regular ones in the network file traffic. This file segregation tool had certain major drawbacks. For example, it had only a single-feature check, meaning it looked only at the file's Shannon's entropy. It gave a high rate of false alarms, it was manually operated, and it had no learning or adapting capabilities.

To overcome the drawbacks of the previous tool, in this research, a Deep Learning-based Malware Arresting Recommendation model is proposed. Four Deep Learning-based Neural Network frameworks are trained and evaluated for selecting one of them as the backbone of the proposed Recommendation model. The deep models that were chosen for this research were a baseline CNN, a three-layer LSTM, BERT-Large, and fine-tuned CodeBERT.

CodeBERT delivered the highest balanced accuracy of 95.4 %, an F1-score of .984, but a large inference time of 13.9 ms per file. As the pre-trained CodeBERT model has immense potential if fine-tuned on larger datasets, it can be used for slower, deeper inspection of the malicious file traffic. CNN followed CodeBERT with a 93.5 % balanced accuracy, 0.988 F1-score and an inference time of only 0.1 ms per file. It was able to identify almost every malware with very little delay. LSTM, on the other hand, delivered a 92.3 % balanced accuracy, 0.957 F1-score and 0.3 ms per file inference time. It was nearly as sharp as CNN but three times slower. However, the BERT-Large model delivered abysmal results of 54.5 % balanced accuracy, 0.314 F1-score and 24.5 ms inference time per file. It was slow and missed half the threats.

Despite the BERT model's unsatisfactory performance, this Large Transformer model is not totally unsuitable. Being pre-trained on English language data (unlike CodeBERT, which is pre-trained on both English language and programming language data), it needs more training epochs to be fine-tuned for malicious file classification. It is hoped that BERT will deliver higher balanced accuracy on larger datasets and with more training epochs.

As both speed and precision of the prediction are important, a fully automated Hybrid CNN + CodeBERT-based Malware Arresting Recommendation model is proposed for use in the Naval Ships and Submarines. This model will be able to flag malicious files with a latency of only 0.1 ms by the fast CNN head. The flagged files are passed to the CodeBERT body for a deeper inspection. CodeBERT reads the code-like structure of the file and generates the final prediction. Safe files are released, and the malicious ones are locked in quarantine. This two-step flow keeps 95 % of traffic moving at full speed while still catching stealthy malware.

## 9. Acknowledgements

The authors would like to thank their co-author and mentor, Commander Rakesh Kumar Gosain, for providing immense support and valuable guidance in completing this conference paper. His thirty years of versatile Naval

experience have been a huge stepping stone towards the successful accomplishment of the project, which otherwise would not have been possible. The author would also like to express gratitude towards their family and friends for providing immeasurable encouragement and motivation. Last but not least, a token of appreciation to the IMarEST team for giving this opportunity to showcase this research.

## 10. References

Gosain, A., & Gosain, RK. (2022). Preliminary investigation method for segregating malware-encrypted files from the regular traffic. Institute of Marine Engineering, Science and Technology (IMarEST). https://doi.org/10.24868/10650

GitHub 2022, 'DikeDataset - labeled dataset containing benign and malicious PE and OLE files', GitHub, viewed 20 May 2025, <https://github.com/iosifache/DikeDataset>.

Shannon, C.E., 1948. A mathematical theory of communication. *The Bell system technical journal*, *27*(3), pp.379-423.

Wang, W., Zhu, M., Zeng, X., Ye, X. and Sheng, Y., 2017, January. Malware traffic classification using convolutional neural network for representation learning. In *2017 International conference on information networking (ICOIN)* (pp. 712-717). IEEE.

Cao, X., Luo, Q. and Wu, P., 2022. Filter-GAN: imbalanced malicious traffic classification based on generative adversarial networks with filter. *Mathematics*, *10*(19), p.3482.

Kalash, M., Rochan, M., Mohammed, N., Bruce, N.D., Wang, Y. and Iqbal, F., 2018, February. Malware classification with deep convolutional neural networks. In *2018 9th IFIP international conference on new technologies, mobility and security (NTMS)* (pp. 1-5). IEEE.

Hwang, R.H., Peng, M.C., Nguyen, V.L. and Chang, Y.L., 2019. An LSTM-based deep learning approach for classifying malicious traffic at the packet level. *Applied Sciences*, *9*(16), p.3414.

Ullah, F., Ullah, S., Srivastava, G. and Lin, J.C.W., 2024. IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. *Digital Communications and Networks*, *10*(1), pp.190-204.

Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D. and Zhou, M., 2020. Codebert: A pre-trained model for programming and natural languages. *arXiv preprint arXiv:2002.08155*.

Zhang, H., Lu, S., Li, Z., Jin, Z., Ma, L., Liu, Y. and Li, G., 2024. CodeBERT-Attack: Adversarial attack against source code deep learning models via pre-trained model. *Journal of Software: Evolution and Process*, *36*(3), p.e2571.

Ferrag, M.A., Ndhlovu, M., Tihanyi, N., Cordeiro, L.C., Debbah, M., Lestable, T. and Thandi, N.S., 2024. Revolutionising cyber threat detection with large language models: A privacy-preserving bert-based lightweight model for iot/iiot devices. *IEEe Access*, *12*, pp.23733-23750.

Rahman, M.A., Francia III, G., Shahriar, H., El-Sheikh, E. and Ahamed, S.I., A Novel Approach to Fine-tune BERT using Non-Text Features for Enhanced Ransomware Detection.

Han, L., Zeng, X. and Song, L., 2020. A novel transfer learning based on albert for malicious network traffic classification. *International Journal of Innovative Computing, Information and Control*, *16*(6), pp.2103-2119