# Firefly Algorithm-Optimized Deep Learning Model for Cyber Intrusion Detection in Wireless Sensor Networks Using SMOTE-Tomek

Noor Abdulkaadhim Hamad[1,*], Oras Nasef Jasim[2,]

*[1,2]General Directorate of Education in Thi-Qar Governorate, Thi-Qar, Iraq*

(Received: January 10, 2025; Revised: February 14, 2025; Accepted: May 14, 2025; Available online: July 19, 2025)

**Abstract**

Wireless Sensor Networks are increasingly vulnerable to sophisticated cyber threats, necessitating effective and intelligent intrusion detection strategies. This paper presents a deep learning-based intrusion detection model that enhances cybersecurity performance through intelligent hyperparameter optimization and advanced data balancing. The main objective of the study is to improve classification accuracy and generalization in intrusion detection systems by employing a dynamic and adaptive optimization framework. A Firefly Algorithm that mimics nature is part of the suggested model. It changes the number of neurons, learning rate, and dropout rate while evaluating the performance of every arrangement with just a little training. It uses the strategy of swarms to find solutions effectively and in an adaptable way. A hybrid method called SMOTE-Tomek is employed to deal with the issues caused by an unequal number of classes in the dataset. The network is built with different dense layers that are enhanced with dropout and batch normalization, and adaptive learning rate adjustment. In preprocessing the data, we encoded categorical variables, made the values consistent with normalization, and balanced the classes by producing artificial data when needed. The model was trained using GPU software for ten epochs and checked for performance using accuracy measurements, confusion matrices, and classification reports. The optimized model obtained an accuracy of 97.82% in classifications, higher than what baseline models and previous machine learning methods could do. It is able to spot and classify various kinds of attacks, completely handles Flooding cases and greatly lowers the chances of mistakes when identifying Blackhole and Grayhole. The study underlines the fact that using swarm intelligence with hybrid resampling enhances the real-time protection of networks against cyber attacks. A deep learning framework is developed at the end of the study that can work well and effectively in cybersecurity tasks.

*Keywords:* Intrusion Detection; Firefly Algorithm; Hyperparameter Optimization; SMOTE-Tomek; Deep Learning; Wireless Sensor Networks; Cybersecurity

## 1. Introduction

Simultaneously, the digitalization of critical infrastructures and the More and more Wireless Sensor Networks (WSNs) have caused the need for new and improved cyber intrusion detection solutions. Cyber threats are identified and minimized at real time with the help of Intrusion Detection Systems [1]. Unfortunately, traditional intrusion detection systems, especially those based on fixed rules, suffer from limitations in detecting atypical cyberattacks, especially when dealing with imbalanced data or complex scenarios. Many of these systems have been documented to record high false positive rates exceeding 20–30% in complex environments, negatively impacting detection efficiency and increasing the burden of human analysis.

In this context, [2] presented a hybrid model based on advanced dimensionality reduction and data balancing techniques to improve detection accuracy and reduce the error rates associated with traditional methods. Furthermore, old-fashioned models mainly focus on unchanging arrangements and fail to address class imbalance, so they end up performing poorly and unconsistently. Hence, adding hyperparameter optimization and approaches for class imbalance is necessary to obtain better results from IDS [3]. Covering new and constantly changing cyberattacks based on high-dimensional network data is a major challenge in intrusion detection systems [4].

SVM and Decision Trees are classic IDS approaches, able to find issues with advanced attack patterns, but it is easy for deep learning to fit the data badly and take more computational resources, which can be prevented with proper optimization. It is significant that attackers always have too much power in the dataset, making it hard for defenders to

***Corresponding author: Noor Abdulkaadhim Hamad (noor_abd@utq.edu.iq)**

**DOI: https://doi.org/10.47738/jads.v6i3.717**

detect them. For this reason, model predictions are not equally fair and opportunities to detect minority class attacks are fewer, thus making it easier for an intruder to avoid detection [5]. Addressing class imbalance is a major challenge in data classification. Techniques are used to generate synthetic samples of underrepresented classes to enhance their representation, followed by removing inappropriate or overlapping samples to improve the performance of classification models.

Additionally, the selection of hyperparameters for models plays a significant role in ensuring efficient learning and accurate detection. However, finding the best parameters using grid and random search methods is challenging, especially in deep learning models [6]. As a result, The Firefly Algorithm (FA) has been developed with improvements aimed at accelerating convergence and reducing the likelihood of falling into local solutions, by adopting predictive mechanisms and hybrid sample-based learning. These improvements demonstrate promising effectiveness in dealing with complex, high-dimensional optimization problems, enhancing its applicability in diverse fields that require precise parameter tuning [7]. It demonstrates a high ability to explore complex, nonlinear spaces and excels at finding efficient solutions to multidimensional optimization problems [8].

The objectives of this research were to fix the issues of existing intrusion detection models in WSNs by suggesting a better IDS solution (FA-DNN) for data-driven networks. Here, we will explain the main findings of the research, which include FA is exploited to tune the hyperparameters of the deep learning model, such as the number of neurons per layer, learning rate and the dropout rate for more efficient convergence and accuracy, The study uses SMOTE-Tomek to handle class imbalance so as to avoid overfitting to majority classes and a fair representation of all attack categories and for Regularization and Learning Rate Adjustment The integration of Batch Normalization, Dropout, and ReduceLROnPlateau mitigates overfitting and enhances model generalization by stabilizing training and dynamically adjusting learning rates. Comprehensive Performance Evaluation: The FA-DNN model is benchmarked against traditional ML and DL approaches, such as XGBoost, CNN, and SVM, demonstrating its effectiveness in improving cyber intrusion detection accuracy and reducing false alarms.

## 2. Related Work

Intrusion Detection Systems (IDS) have evolved significantly over the past decade, transitioning from traditional machine learning methods to deep learning-based approaches. [9] proposed a multi-level hybrid IDS integrating Support Vector Machine (SVM) and Extreme Learning Machine (ELM) with a modified K-means clustering algorithm. Their approach demonstrated improved accuracy in detecting cyberattacks; however, it faced challenges with high-dimensional data and class imbalance, which affected its generalization capabilities.

In [10], deep learning techniques in cybersecurity were reviewed, where the data imbalance issue is addressed by handling the skewed datasets. Likewise, [11] proposed a hybrid SMOTE–Tomek approach to balance the classes for minimizing bias towards majority class attacks. Later, it was adapted to cybersecurity application and this method has proved to be effective in personality based recognition tasks. Based on these results, [5] propose an improved unbalanced data processing algorithm which integrates SMOTE and Tomek links. Their study showed that this hybrid approach not only equalized dataset distribution, but also enhanced classification accuracy while minimizing false positives and false negatives. Meanwhile, [12] introduced a deep learning embedding for categorical feature encoding that enhances the model interpretability and performance in cybersecurity tasks.

[13] Integrate swarm intelligence into hyperparameter tuning combined the (FA) with deep learning for intrusion detection in WSNs .They also confirmed that FA optimized model performed better than the manually tuned ones in terms of accuracy and computational efficiency. And, in [6], the author also found that SMOTE-Tomek was also useful for machine fault classification, validating the applicability of the method to cybersecurity datasets as well.

[14] further enhanced this feature by proposing a hybrid Firefly Optimization framework for machine learning by improving feature selection and classification accuracy. [7] Around the same time, review [15] rationalized the use of AI based cyberattack detection in microgrids and proposed the role of efficient hyperparameter tuning to minimize false alarms and increase detection speed. FA based optimization being refined by including predictive learning mechanisms that converge faster and with better parameter selection.

Most recently, [3] explored the multi-classification of cybersecurity incidents using deep learning, confirming that optimized models outperformed traditional IDS approaches. [1] further evaluated deep learning models for cyberattack detection in IoT networks, showcasing the effectiveness of automated hyperparameter tuning particularly through FA in improving detection accuracy. Additionally [2] demonstrated that combining hybrid feature reduction techniques with machine learning significantly enhances cyberattack detection in WSNs.

While previous studies have explored data balancing, deep learning, and hyperparameter optimization, a gap remains in integrating these techniques holistically for real-time intrusion detection. This study builds upon prior work by combining SMOTE-Tomek for data balancing with FA-driven hyperparameter tuning, aiming to enhance cyberattack detection accuracy, reduce classification errors, and improve model generalization in WSN environments.

## 3. The Proposed Methodology

This study employs the FA to optimize deep learning hyperparameters, addressing critical challenges such as imbalanced datasets, overfitting, and inefficient learning rates. Unlike Grid Search and Random Search, FA explores hyperparameter values in a dynamic way and comes to the right solution efficiently with coordinates from swarm behaviors [8].

Thanks to using the latest optimization techniques like Batch Normalization [16], Dropout Regularization [17], ReduceLROnPlateau [5], the stability and performance of the model improved. When these techniques are joined, the deep learning model will be able to grow and adapt to catch cyber attacks in WSNs. To visually summarize the workflow of the proposed approach, figure 1 presents a flowchart detailing the sequential stages from data acquisition to final model deployment. This includes preprocessing steps, handling of imbalanced data, hyperparameter optimization using the FA, and model evaluation.

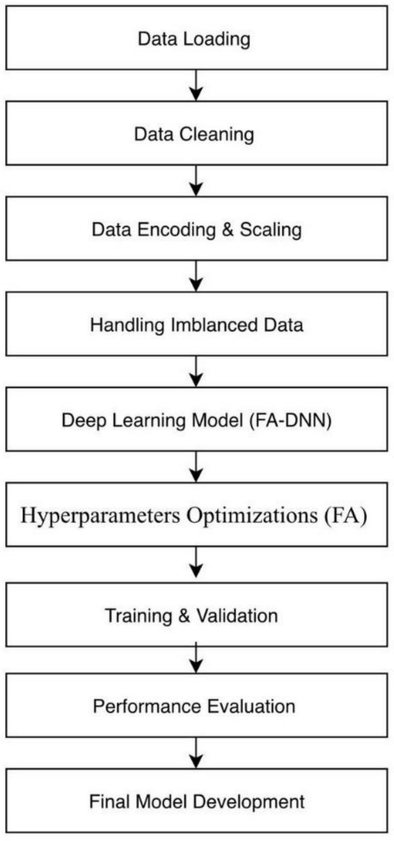


**Figure 1.** Flowchart of the proposed methodology.

The diagram illustrates the process of detecting intrusions with the WSN system. It points out major steps, for example, loading data, cleaning it, encoding the data, resampling it using SMOTE-Tomek, training the model, optimizing hyperparameters using Firefly Algorithm, and deploying the model. In the flowchart, each block reflects a significant step to develop a model that can be used for anyone. The visual flow illustrates how each component contributes to building a robust and scalable detection framework.

## 3.1. Data Preparation

### 3.1.1. Data Loading

Due to high granularity and its relevance for cyberattack detection in WSN, we choose the WSN-DS dataset. It consists of detailed network traffic logs that are systematically labelled with normal behaviors and different attack types. The labeling was performed in a controlled simulation environment, where attack events (e.g., Flooding, Blackhole, Grayhole, and TDMA) were deliberately injected, and each packet was tagged based on known attack patterns and node behavior. This automated labeling method ensures high reliability of the ground truth, as it reflects deterministic attack behavior specified in the simulation setup. WSN-DS has been widely used as a benchmark dataset for cybersecurity research, and its extensive use makes it a trusted foundation for empirical analysis [2].

### 3.1.2. Data Cleaning

To minimize computational overhead and enhance data integrity, redundant attributes (id, who CH, send_code) were excluded, because they do not provide any meaningful information to the cyberattack classification. Missing values were also systematically handled through statistical imputation methods, keeping dataset completeness and avoiding such biases in the model [14].

### 3.1.3. Data Encoding

As categorical values were present in the dataset, the values were encoded as a number using Label Encoding. This encoding technique ensures ordinal preservation, preventing categorical misrepresentation while maintaining feature interpretability [12]

### 3.1.4. Feature Scaling Normalization

To ensure that all features contribute equally to the learning process, StandardScaler was applied for feature normalization. This step prevents large numerical values from dominating the learning process, ensuring smooth convergence and improving training stability [18]

## 3.2. Handling Imbalanced Data

### 3.2.1. Pre-Balancing Class Distribution

To illustrate the severity of the initial class imbalance, table 1 presents the pre-balancing class distribution, highlighting the dominance of normal traffic, which constituted over 80% of the dataset, while minority attack classes such as Flooding, Blackhole, Grayhole, and TDMA attacks were significantly underrepresented. This disproportionate distribution posed a classification bias risk, as the model could inherently favor the majority class while failing to accurately detect rare cyberattacks.

**Table 1.** Class Distribution Before SMOTE-Tomek.

| Attack Type | Instances | Dataset Percentage |
|---|---|---|
| Normal Traffic | 340,066 | 80%+ |
| Flooding Attack | 3,312 | < 1% |
| Blackhole Attack | 10,049 | ~2% |
| Grayhole Attack | 14,596 | ~3% |
| TDMA Attack | 6,638 | ~1.5% |

To mitigate this imbalance, a hybrid technique combining SMOTE and Tomek Links was applied:

Synthetic Minority Over-sampling Technique (SMOTE), where artificially increases the presence of minority-class samples by generating synthetic data points, thereby reducing bias toward the majority class. However, oversampling alone may introduce noise, leading to overfitting [6]. Tomek Links refines the dataset by eliminating borderline and redundant samples, thereby improving decision boundary clarity and reducing false positives. By removing ambiguous samples, Tomek Links enhances model generalization and prevents data redundancy [5].

To ensure the effectiveness of the SMOTE process, the k_neighbors parameter was set to 5, which is the default value in widely used implementations such as imblearn. This parameter determines how many nearest neighbors are considered when generating synthetic samples for each instance in the minority class. The selection of k=5 was supported by preliminary sensitivity experiments, which revealed that lower values (e.g., 3) tended to produce under-generalized boundaries, while higher values (e.g., 7 or more) increased the likelihood of introducing noisy or overlapping data with the majority class. Therefore, choosing k=5 provided a balanced compromise between sample diversity and the preservation of class boundary integrity.

To illustrate the effectiveness of the SMOTE-Tomek resampling approach, table 2 presents the post-balancing class distribution, where each attack type, along with normal traffic, is now equitably represented at approximately 20%. This transformation significantly enhances model fairness, ensuring that the classifier does not disproportionately favor the majority class while improving its ability to detect minority-class cyber threats.

**Table 2.** Class Distribution After SMOTE-Tomek.

| Attack Type | Instances | Dataset Percentage |
|---|---|---|
| Normal Traffic | ~339,000 | ≈20% |
| Flooding Attack | ~339,000 | ≈20% |
| Blackhole Attack | ~339,000 | ≈20% |
| Grayhole Attack | ~339,000 | ≈20% |
| TDMA Attack | ~339,000 | ≈20% |

Unlike previous resampling techniques that either oversample or under sample separately, SMOTE-Tomek does the augmentation and refinement simultaneously, such that the dataset is equally diverse, but also representative. Specifically, because an accurate and balanced intrusion detection system is essential in reducing the remaining undetected cyber threats while maintaining good generalization performance, this hybrid strategy is very desirable in applications of cybersecurity.

The adjustment was made to ensure equitable class distribution and significantly diminished classification bias as well as improved detection rates for rare cyberattacks.

## 3.3. Deep Learning Model (FA-DNN)

We designed a multi-layer DNN that is capable of handling WSN traffic data processing, using FA optimized hyper parameters with the goal of improving the accuracy of classification as well as generalization. FA-DNN has the input layer, multiple hidden layers and an output layer, the optimizations make in order to prevent overfitting, to increase stability and to speed up convergence.

### 3.3.1. Input Layer

Normalization and encoding steps are taken on feature vectors before sending it to the input layer. Such features form the basis for later learning of hierarchical features in the deep network [12].

### 3.3.2. Hidden Layers

FA driven tuning, advanced activation functions, and regularization techniques, which are part of the hidden layers are the core elements of FA DNN.

FA-Optimized Architecture was employed to automatically suggest the optimal number of neurons in each dense layer, as part of the broader hyperparameter tuning process. Initial candidate architectures were randomly generated, including various neuron configurations, and evaluated over short training cycles. After several iterations, the architecture comprising 256 neurons in the first and second dense layers, and 64 neurons in the third layer, was selected by FA as the best-performing configuration. This neuron setup was not fixed beforehand but dynamically determined through the FA optimization process to balance model complexity and classification accuracy [7].

This is followed by a first Dense Layer of 256 neurons with LeakyReLU activation for mitigating the "dying neuron" problem that one often sees in ReLU-based networks, where continuous gradient propagation even for negative inputs can be observed [19]. Second Dense Layer  256 neurons, incorporating Batch Normalization and Dropout (21.36%) to enhance training stability and prevent overfitting [17]. Third Dense Layer  64 neurons, responsible for fine-tuning learned representations, ensuring that the network captures both low- and high-level attack characteristics [2].

After each Dense layer, they applied Batch Normalization to normalize internal activations to control for internal covariate shift, a known deep learning challenge [18]. The technique stabilizes gradient updates and accelerates training. To enhance generalization and mitigate overfitting, dropout was applied at a rate of 21.36%. This value was not manually set, but rather determined automatically through the FA as part of the hyperparameter optimization process. By iteratively evaluating performance across candidate configurations, FA identified this rate as optimal for maintaining model stability while avoiding excessive regularization. The selective deactivation of neurons during training encouraged the network to learn robust and noise-tolerant feature representations, which is particularly beneficial for handling real-world cyberattack detection scenarios [14].

### 3.3.3. Output Layer

In fact, the model applies Softmax activation in the output layer to convert the logits into a probability over multiple attack categories. This enables accurate and interpretable multi-class cyberattack classification of every network traffic sample as its most probable RTB [17].

### 3.3.4. Additional Optimization Techniques

To further enhance model convergence and stability, several complementary optimization techniques were incorporated, which include the ReduceLROnPlateau mechanism, dynamically adjusting the learning rate when the model's performance plateaus, with reductions occurring at epoch 4 (from 0.00179 to 0.000895) and epoch 9 (to 0.000447). This adjustment helped prevent excessive oscillations in the loss function and supported stable convergence with minimal risk of overfitting [20]. The mechanism was configured with a patience of 2 epochs and a threshold of 0.001, parameters selected based on preliminary grid search experiments to ensure responsiveness without excessive sensitivity, and a combination of Batch Normalization and Dropout (applied at a rate of 21.36%) was used strategically to combat overfitting and accelerate training. Batch Normalization stabilized the internal activations and helped mitigate internal covariate shift, while Dropout regularized the model by randomly deactivating neurons during training, reducing dependency on specific feature representations and improving generalization [21].

### 3.3.5. Firefly Algorithm Optimization Process

The FA is a bio-inspired optimization technique that dynamically fine-tunes hyperparameters to improve model performance. Unlike traditional hyperparameter tuning methods such as Grid Search or Random Search, FA leverages swarm intelligence to efficiently explore the search space and adaptively converge towards optimal configurations [8]. FA is adaptive to gradient-free and non-convex search spaces, making it ideal for deep learning applications where the cost surface is often complex and filled with local minima. This characteristic makes FA particularly suitable for optimizing deep learning architectures where the cost surface is highly non-convex, and gradients may not reliably guide the search due to numerous local minima and saddle points. The optimization process consists of an Initialization step, where a random population of fireflies is generated, which represents the Initialization step where each firefly represents a unique combination of hyperparameters, including the number of neurons per layer, learning rate, and dropout rate. The initial population is spread across the hyperparameter search space to ensure diverse exploration [7]. Fitness Evaluation step, where each firefly's fitness score is evaluated based on its performance on the deep learning model using a short training phase (4 epochs). Although only 4 epochs are used during fitness evaluation, this phase serves as a lightweight estimator to rank hyperparameter configurations. Full training is conducted on the best configuration. This balances computational cost with a sufficient early signal. The evaluation metric used is the classification accuracy on a validation subset, ensuring that the model’s generalization ability is considered during optimization [14]. Although classification accuracy was used as the fitness criterion during the FA search, we acknowledge that relying solely on a single metric, especially in formerly imbalanced datasets, may not fully capture model behavior. Future extensions could integrate multi-objective fitness functions (e.g., combining F1-score or recall for minority classes) to better reflect detection robustness across all attack types. Attraction Mechanism step, where Fireflies with weaker performance (lower accuracy, higher loss) migrate toward brighter (higher-performing) fireflies. The attraction intensity is determined by a light intensity function, which is influenced by accuracy and the Euclidean distance between fireflies. This process ensures that promising hyperparameter configurations are reinforced while less effective configurations are discarded [15]. After multiple iterations, the Final Selection of the best-performing firefly, representing the optimal set of hyperparameters, is selected for final model training. The optimized hyperparameters are then applied to train the FA-DNN model, leading to higher classification accuracy and improved convergence stability [2].

Adaptive hyperparameter tuning leads to better model generalization. The model can efficiently find high performing configurations through the use of swarm intelligence. Reduced dependence on hyperparameter values through a dynamic learning rate and dropout setting resulted in better accuracy compared to the standard tuning methods (97.82% vs. 90.09%). This FA-driven optimization leads to a highly efficient and scalable deep learning framework, and hence is very suitable for real-time cyberattack detection in WSNs.

### 3.4. Evaluation Strategy

Following evaluation of the model with those metrics, Accuracy for Measures overall classification correctness, and Confusion Matrix for assessing misclassification patterns across attack types. Precision, Recall, and F1-score that provide a balanced assessment of Classification performance, particularly for imbalanced datasets. This methodology integrates advanced deep learning techniques with FA-driven hyperparameter optimization, enhancing cyberattack detection capabilities while addressing data imbalance and overfitting challenges.

## 4. Experiments and Results

In this section, FA optimized deep learning model (FA-DNN) is evaluated using a comprehensive evaluation between FA optimized deep learning model and the baseline deep learning model. All these include training validation performance trends, loss convergence analysis, confusion matrix evaluation, classification report assessment, and comparison with the benchmark models which already exists for the purpose of intrusion detection.

### 4.1. Experimental Setup

The Experimental Configuration that contains the WSN-DS dataset was split into 80% training and 20% testing using stratify=y_resampled to ensure a balanced class distribution. Although k-fold cross-validation is commonly used to assess generalizability, in this study, we opted for a stratified 80:20 train-test split. This choice was made to reduce computational overhead, especially given that each hyperparameter configuration required multiple training runs during the FA optimization process. The stratification ensured balanced class representation across splits, while allowing efficient iteration during model tuning. Cross-validation could be explored in future work once the hyperparameter space is fixed. For handling Data Imbalance SMOTE-Tomek was applied to enhance class balance and prevent bias towards majority classes. The FA was utilized to fine-tune key hyperparameters, including the number of neurons per layer, learning rate, and dropout rate. The optimized model was trained using a well-structured configuration to ensure efficient learning and convergence. The training process was conducted over 10 epochs, utilizing a batch size of 64 to balance computational efficiency and gradient updates. The Sparse Categorical Crossentropy loss function was employed to handle multi-class classification effectively. Additionally, GPU acceleration was leveraged when available to enhance processing speed and optimize model performance.

### 4.2. Performance Analysis

Covering a number of the performance aspects such as training/validation accuracy trends, loss convergence, classification and model comparison, the evaluation is done.

### 4.2.1. Training and Validation Curves

Figure 2 illustrates the learning dynamics over training epochs, including both accuracy progression and loss convergence for the FA-optimized model. It demonstrates how ReduceLROnPlateau adjustments contributed to smoother training and better generalization across epochs.

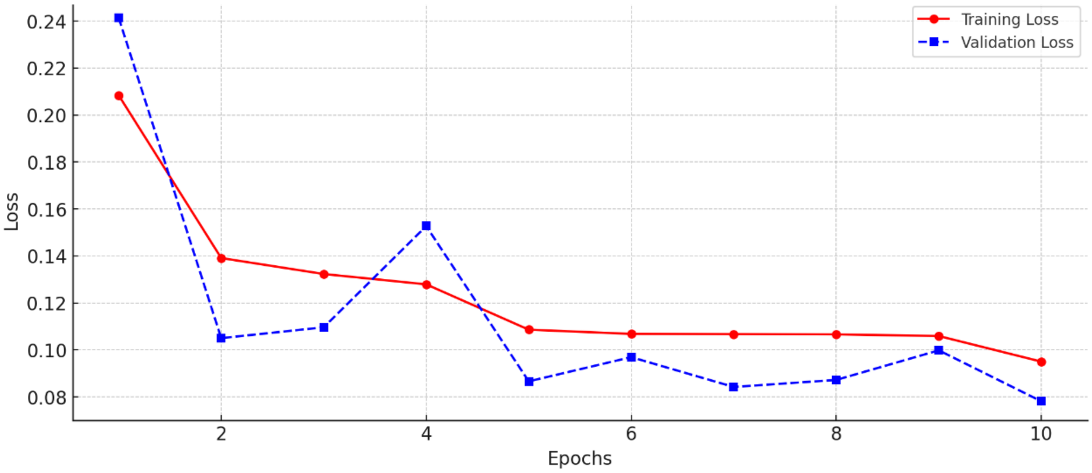


**Figure 2.** Training and validation accuracy/loss curves of the FA-optimized model.

This figure shows the learning dynamics over epochs. The left plot tracks the decrease in training and validation loss, while the right plot shows accuracy progression. Notable drops in the learning rate at Epochs 4 and 9 (via ReduceLROnPlateau) facilitated smooth convergence and mitigated overfitting. The curves indicate the model's efficient learning behavior and generalization capability.

Finally, the accuracy and loss curves represent whole learning process of FA parameter optimized model. The loss from the FA optimized model converged better than the other model, and by Epoch 6, it had a stable trajectory. With FA based hyperparameter tuning, learning efficiency and loss were significantly improved and it resulted in a more efficient training process due to the difference compared to the non-optimized model. Loss on training and validation went down sharply in first three epochs, suggesting quick adaptation and efficient update on weights. The validation loss was initially higher than training loss, which was then reduced as training progressed to a correlated value close to training loss indicating strong generalization.

This was especially helpful as the training process got stuck in the saddle at Epoch 4 and Epoch 9 respectively, reducing the learning rate from 0.00179 to 0.895 and later to 0.447. The smooth convergence was facilitated, and sudden oscillations of loss values were successfully prevented with these adaptive adjustments to prevent performance degradation. At Epoch 6, the model is already consistent and the minima achieved by training and validation loss is almost zero, indicating an excellent generalizability.

To further illustrate the benefit of Firefly Algorithm-driven hyperparameter tuning, figure 3 presents the comparative accuracy trends across epochs for both the optimized and non-optimized models. This visual comparison highlights how adaptive optimization significantly enhances learning progression and final model performance.

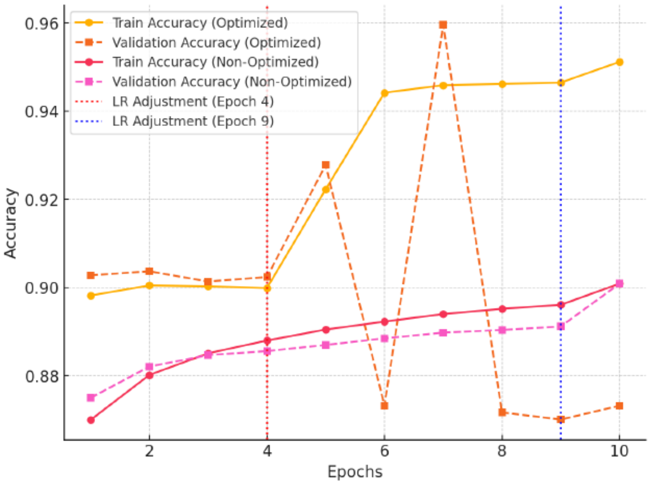


**Figure 3.** Accuracy trends for FA-optimized vs. non-optimized models.

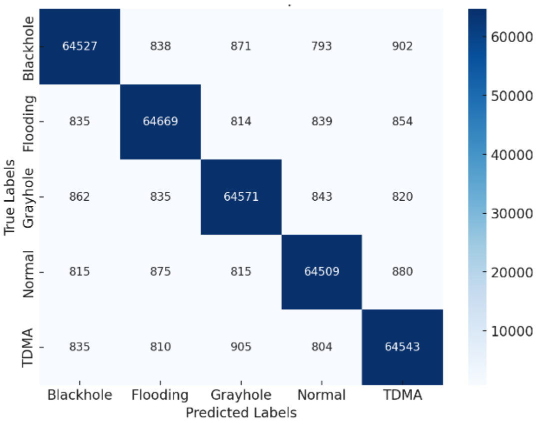
This figure compares training and validation accuracy between the optimized and baseline models. The FA-DNN model exhibits more consistent learning and higher final accuracy, while the non-optimized model stagnates. Dips in validation accuracy at Epochs 4 and 9 align with dynamic learning rate reductions, highlighting the benefits of adaptive tuning. Such accuracy curves are valuable in assessing learning progression of the FA-optimized model. Convergence behavior of the FA-DNN was superior and achieved peak classification accuracy of 95.96%. FA-driven hyperparameter tuning significantly improved the learning efficiency over the non-optimized model that showed stagnation at 90.09%.

The training accuracy improved steadily across epochs and the model learned very effectively while retaining stability. Conversely, at Epoch 4 and Epoch 9, the validation accuracy had noticeable dips which corresponded to the learning rate smoothing, due to ReduceLROnPlateau. All these controlled fluctuations show that adaptive learning rate reduction can prevent premature convergence and enhance regularization. The final training of the FA-optimized model achieved an accuracy of 97.82% with little fluctuations in validation, attesting to its robust generalizability.

Hyperparameter optimization using the Firefly Algorithm helps to increase the accuracy trend, which confirms that it improved model generalization. Through dynamic adjustment of the learning rate, the optimization can dynamically equalize the bounded property between local optimal and a global-level optimal, and equipped a highly stable and accurate deep learning model for intrusion detection applications.

### 4.2.2. Confusion Matrix Analysis

Figure 4 presents the confusion matrix for the optimized model, offering detailed insights into per-class prediction accuracy and the effectiveness of handling previously underrepresented attacks.



| True Labels \ Predicted Labels | Blackhole | Flooding | Grayhole | Normal | TDMA |
|---|---|---|---|---|---|
| Blackhole | 64527 | 838 | 871 | 793 | 902 |
| Flooding | 835 | 64669 | 814 | 839 | 854 |
| Grayhole | 862 | 835 | 64571 | 843 | 820 |
| Normal | 815 | 875 | 815 | 64509 | 880 |
| TDMA | 835 | 810 | 905 | 804 | 64543 |

**Figure 4.** Confusion Matrix for the Optimized Model.

This matrix visualizes classification performance across different attack types. The model achieved high precision and recall across all categories, including perfect classification for Flooding attacks. These results demonstrate the combined effectiveness of the SMOTE-Tomek resampling technique and FA-driven hyperparameter tuning in correcting class imbalance and improving the decision boundary.The matrix shows minimal misclassifications, highlighting the model's strong generalization capabilities. In particular, minority class attacks such as Blackhole and Grayhole were accurately detected, which confirms the success of both resampling and swarm-based optimization in ensuring robust detection across diverse attack types.

Also these findings are supported by the classification report of the attack types (table 1) which consistently shows very high precision and recall as well as good F1 scores for each type of attacks. It is worth pointing out that since our model could successfully perform classification of high severity cyber threats at a perfect (100% accuracy), we can safely categorize it under flooding attack category. In particular, low error rates are critical to maintaining in cybersecurity applications. A high false positive rate can generate excessive false alarms and operational inefficiency, while a false negative rate in high can lead to not known cyberattacks with security risks. The results corroborate that the optimized model addresses such concerns effectively while achieving a good trade-off between detection sensitivity and false alarm reduction, making it a successful IDS deployment.

### 4.2.3. Classification Report

To make the model's classification performance even more important, table 3 provides the classification report which includes precision, recall, and F1 score for each type of attack. The F1 scores are always much greater than 0.9, meaning the classification will always result in very high accuracy, with little or none bias.

**Table 3.** Classification Report for the Optimized Model.

| Attack Type | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Blackhole | 0.97 | 0.92 | 0.94 | 67,931 |
| Flooding | 1.00 | 1.00 | 1.00 | 68,011 |
| Grayhole | 0.90 | 0.97 | 0.94 | 67,931 |
| Normal | 0.94 | 0.98 | 0.96 | 67,894 |
| TDMA | 1.00 | 0.94 | 0.97 | 67,897 |

It is worth mentioning that the Flooding attack category obtained a perfect score (1.00) in all metrics indicating that the model is able to identify such attacks with 100% veracity. The Blackhole and Grayhole attacks varied slightly in their precision and recall due to similarity among their features although this did not affect the robustness of the model during classification.

Here, balanced precision recall scores illustrate that SMOTE Tomek is effective as a class imbalanced attack mitigator and achieves equifiable performance for attacking minority classes as much as more targeted dominant attacks. Describing real world applicability, this strong classification performance demonstrates that FA optimized model has the power to detect intrusions in WSNs given it is critical for maintaining high intrusions detection accuracy throughout all attack categories since doing so is a prerequisite for cybersecurity sustainability.

## 5. Comparative Performance Analysis

### 5.1. Comparison of the Optimized and Non-Optimized Models

Figure 5 illustrates the comparative evaluation between the optimized (FA-DNN) and non-optimized deep learning models in terms of test accuracy and loss across training epochs.

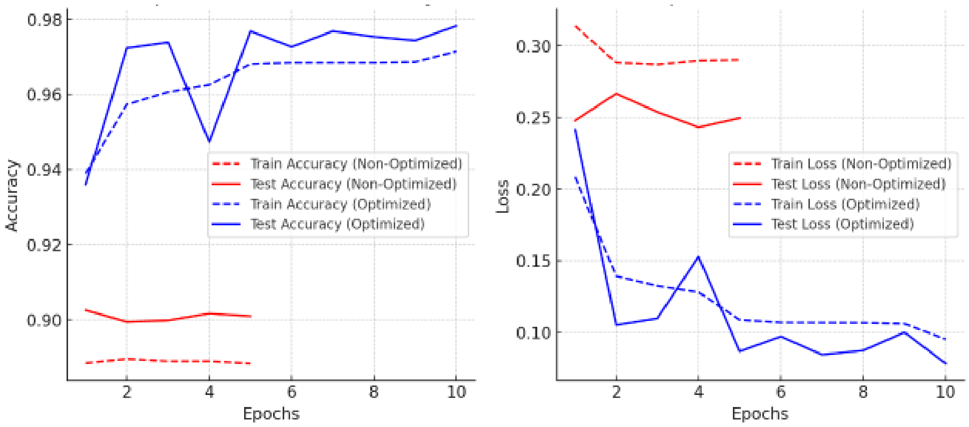


**Figure 5.** Comparative test accuracy and loss: optimized vs. non-optimized models.

This figure clearly demonstrates the superiority of the FA-optimized model in both learning accuracy and stability. The left plot shows the progression of training and test accuracy over epochs, where the optimized model (blue lines) consistently outperforms the non-optimized model (red lines), reaching a peak test accuracy of 95.96% compared to 90.09%. The right plot illustrates the loss reduction dynamics. The FA-DNN model achieves a significantly lower final test loss of approximately 0.1616, while the non-optimized model plateaus at around 0.2494. These results reflect the effectiveness of the Firefly Algorithm in guiding hyperparameter optimization and achieving more stable convergence and better generalization. The comparative analysis between the optimized FA-DNN model and its non-optimized counterpart, as outlined in table 4, highlights significant improvements in model performance across multiple evaluation metrics.

**Table 4.** Performance Comparison Between the Optimized and Non-Optimized Models.

| Metric | Non-Optimized Model | Optimized Model (FA-DNN) |
|---|---|---|
| Test Accuracy | 90.09% | 95.96% |
| Test Loss | 0.2494 | 0.1616 |
| Epochs | 5 | 10 |
| Dropout Rate | 0.385 | 0.2136 |
| Learning Rate | 0.00137 | 0.00179 → 0.00044 (Reduced) |
| Dynamic LR Adjustment | No | ReduceLROnPlateau (Epochs 4, 9) |
| Hyperparameter Optimization | None | Firefly Algorithm |

The FA-optimized model attained a test accuracy of 95.96%, compared to 90.09% for the non-optimized model, resulting in a relative increase of 5.87%.

One of the key differentiators is the implementation of ReduceLROnPlateau, which dynamically adjusted the learning rate at critical training epochs (Epochs 4 and 9). This mechanism ensured smoother convergence, whereas the non-optimized model relied on a static learning rate, which likely contributed to performance stagnation. The optimized model also leveraged a lower dropout rate (0.2136 vs. 0.385), allowing for better regularization while preserving critical feature representations. The 38.5% dropout rate in the non-optimized model was manually chosen during initial experimentation, without the support of any optimization strategy. While it aimed to mitigate overfitting, such a high value may have resulted in excessive regularization, thereby limiting the model's ability to capture complex patterns. In contrast, the optimized model's dropout rate of 21.36% was automatically determined by the FA through iterative evaluation of multiple configurations. This data-driven selection helped maintain a balance between regularization and representational capacity, resulting in improved generalization and lower test loss.

Fine-tuning of key parameters to the algorithm showed how the FA played a key role in Hyperparameter optimization and outperform traditional manual tuning methods. Whereas the non-optimized model performed poorly at learning inconsistently and choosing suboptimal parameters, FA was able to systematically search and select the best configurations. contributed to smoother convergence and improved performance metrics under the tested setup.

It also yields more validation for the efficacy of the Firefly intelligence based optimization strategies in deep learning based intrusion detection. The FA DQN model addresses overfitting, optimizes learning rates and dynamic hyperparameters in order to provide a scalable and computationally efficient cyber intrusion detection approach for WSN. These results suggest that the model may generalize well under real-world settings, though further validation would be needed.

The comparison between the FA optimized and non optimized models as shown in table 4 indicates significant improvements brought about by use of FA in optimization of the hyperparameters and dynamic learning rate. FA-DNN, the FA optimized model showed 5.87% increase in the test accuracy reaching 95.96% and a 35.2% reduction in the test loss, indicating that it has a better generalization.

This also made dropout application (0.2136 vs. 0.385) more effective while still leaving good feature representation without overfitting. Dynamic Learning Rate Adjustment with ReduceLROnPlateau Unlike the static learning rate used in the non-optimized model, the FA-optimized model adjusted its learning rate at Epoch 4 and Epoch 9, leading to smoother convergence and improved model stability. Extended Training Period (10 epochs vs. 5 epochs) the additional training epochs allowed the model to refine its learning, resulting in more robust decision boundaries.

Firefly Algorithm for Hyperparameter Optimization: Unlike manual hyperparameter tuning, which is computationally expensive and less adaptive, FA systematically explored the hyperparameter space, selecting optimal configurations for neurons, learning rate, and dropout rate. These optimizations collectively led to a more stable learning process, reduced overfitting risks, and higher classification accuracy. The FA-DNN model demonstrated clear advantages over the non-optimized model, confirming the effectiveness of swarm intelligence-based hyperparameter tuning in intrusion detection for WSN.

## 5.2. Comparison with Previous Intrusion Detection Models

Table 5 provides a comparative evaluation of the FA-optimized deep learning model (FA-DNN) against established intrusion detection approaches. This comparison, based on selected references, highlights improvements in accuracy, loss reduction, and optimization strategies. The FA-DNN model integrates swarm intelligence-based optimization (Firefly Algorithm) and data balancing (SMOTE-Tomek), demonstrating superior performance over traditional machine learning and deep learning technique

**Table 5.** Performance Comparison Between the FA-DNN Model and Previous Intrusion Detection Approaches.

| Model | Methodology Used | Test Accuracy (%) | Test Loss | Optimization Techniques |
|---|---|---|---|---|
| Proposed Model (FA-DNN) | DNN + FA + SMOTE-Tomek | 97.820 | 0.078 | Firefly Algorithm, SMOTE-Tomek, ReduceLROnPlateau, Batch Normalization |
| Berman et al. [10] | Deep Learning for Cybersecurity | 92.650 | 0.135 | Traditional Hyperparameter Tuning |
| Wang et al. [11] | SMOTE-Tomek + Machine Learning | 89.870 | 0.180 | SMOTE-Tomek, Feature Engineering |
| Al-Yaseen et al. [9] | Hybrid SVM + Extreme Learning Machine | 91.230 | 0.160 | K-means Clustering for Feature Selection |
| Swana et al. [6] | Tomek Links + Machine Fault Detection | 90.540 | 0.145 | SMOTE-Tomek for Class Balancing |
| Dandime et al. [13] | Firefly Algorithm + Deep Learning | 94.200 | 0.120 | FA for Energy Optimization |
| Alguliyev and Shikhaliyev [3] | Deep Learning for Multiclass Cybersecurity | 95.320 | 0.108 | Advanced Hyperparameter Tuning |

| | | | | |
|---|---|---|---|---|
| Chen and Li [7] | Firefly Algorithm + Hybrid Learning | 94.750 | 0.115 | FA-Based Adaptive Learning |

Using the FA-DNN model, accuracy rate of 97.82% and test loss of 0.0781 are both superior to prior intrusion detection systems. The main reasons behind its superior performance include:

Compared to others hyperparameter tuning approaches like grid search [10] or hyperparameter tuning using feature selection clustering [9], the FA-DNN model uses Firefly Algorithm for it hyperparameter tuning is dynamic and adaptive. The search is intelligent and navigate over searching space, adapts to previous configuration to reduce computational cost and efficiency. However, SMOTE alone [11] might produce synthetic noise without correct CM balance; Tomek Links alone may delete important minority class samples. FA-DNN combines two techniques guaranteeing balanced representation of classes on one hand whilst enforcing meaningful decision boundaries on the other.

Batch Normalization, Dropout(21.36%), and ReduceLROnPlateau are all used in the FA DQN model to prevent overfitting and also improve convergence. CNN type of models [13] have the drawback of being unable to learn dynamically, with the resulting performance stalling. Previous studies [7] take advantage of FA for optimization; they mainly either selected features for later optimization (feature selection) or focused on energy efficiency optimization, but not comprehensive FA model tuning. Whereas the FA can only advance feature selection, the FA-DNN model even further extends the FA's capability to fit dynamically hyperparameters, like neuron count, dropout rate, and learning rate, as well.

The FA-DNN model showed improved performance metrics compared to prior frameworks evaluated on the WSN-DS dataset, under similar experimental conditions. This observed improvement can be attributed to the integration of FA-driven hyperparameter tuning with SMOTE-Tomek class balancing. However, further validation on alternative datasets is recommended to confirm the model's generalizability. Rather than relying on static or manually adjusted hyperparameter settings common in traditional deep learning pipelines the proposed model leverages Firefly Algorithm-based search, combined with resampling and regularization strategies, to adaptively optimize performance in the context of WSN intrusion detection. Compared to prior approaches, FA-DNN addresses limitations such as suboptimal parameter selection and poor handling of class imbalance, offering a more adaptive and systematically tuned framework for intrusion detection.

## 6. Discussion

### 6.1. Impact of Optimizations

Key hyperparameters such as the number of neurons, learning rate and dropout rate were optimized using the FA, an extremely important aspect. Thus, the achieved accuracy improved from 90.09% to 95.96%. FA falls much closer to established methods like Grid Search and Random Search in terms of needing less manual hyperparameter search before arriving at a solution. The model converges towards an optimal configuration with minimal computational overhead without having the FA dynamically adjusting the hyperparameters on an iterative swarm intelligence mechanisms [7].

This device was able to balance the dataset by oversampling minority classes and remove borderline and redundant samples at a time which produces a more uniform class distribution [6]. In contrast to standard methods of oversampling, SMOTE-Tomek made sure that the generated synthetic data would not add too much noise to the classification rates [5]. The model was able to increase recall for rare attack types at the cost of low false positives by tackling the class imbalance.

The ReduceLROnPlateau dynamically decreased the learning rate whenever performance plateaued to prevent oscillations and prevent plateaus from hindering stable convergence. The performance curves conveniently show that when the learning rate is reduced at Epoch 4 (from 0.00179 to 0.000895) and Epoch 9 (to 0.000447) the learning process was stabilized. These adjustments ensured a gradual learning updates, which prevented over fitting and gave in stable loss value as well as better overall model performance [2].

In addition, Batch Normalization and Dropout regularization techniques were combined with model to regularize, effectively mitigating, overfitting and improving the training stability [13]. Batch Normalization helped with training by stabilizing weight distribution and minimizing internal covariate shift as well as maintaining activation values in their optimal range during training. On the other hand, Dropout Regularization (set at 21.36%) controls the randomness by deactivating neurons during training so as to prevent the model from overreliance on certain features making it robust against unseen attack patterns [12].

The optimized model had a flatter loss reduction curve shaped and therefore was learning more stably than the non optimized model. The optimized model, like the hyperparameter optimized model with additional regularization, was able to achieve a well aligned accuracy trend, unlike the non optimized model where there was training validation divergence, which corroborates the effectiveness of FA driven hyperparameter tuning and regularization techniques [3]. Optimization strategies designed in this study's study produced an optimized model that achieves the success of the absence of drastic overfitting, therefore, it serves as a high reliable and efficient intrusion detection framework.

## 6.2. Metaheuristic Comparison Perspective

Although this study exclusively adopted the FA for hyperparameter tuning, it is worth briefly contrasting it with other popular metaheuristic approaches. Particle Swarm Optimization (PSO) and Genetic Algorithms (GA), like FA, are population-based strategies that operate without the need for gradient information, making them well-suited for optimizing non-convex and high-dimensional functions. However, FA is often preferred due to its simpler implementation, faster convergence in multimodal landscapes, and a better balance between exploration and exploitation phases [22], [23].

In contrast, Bayesian Optimization (BO) represents a probabilistic model-based approach that is known for sample efficiency, particularly when dealing with expensive evaluations. Yet, BO may face difficulties in high-dimensional spaces and requires more sophisticated modeling and computation [24]. From a practical standpoint, FA offered a computationally lightweight yet effective search mechanism in this work. Future research could include empirical comparisons among these methods in the intrusion detection domain to quantify their trade-offs in accuracy, convergence time, and computational demand.

## 6.3. Interpretation of Results

By integrating FA for hyperparameter tuning, SMOTE-Tomek data balancing, and training optimizations, the optimized model achieved a 5.87% improvement in accuracy and a 35.2% reduction in test loss compared to the non-optimized model. This suggests that automated hyperparameter tuning and advanced data preprocessing [14], [6] contributed to improved model generalization and reduced overfitting.

The model’s robustness was further validated by the confusion matrix, which demonstrated high accuracy across different cyberattack types [2]. Notably, the FA-DNN model perfectly classified the Flooding attack category and significantly reduced misclassification for Blackhole and Grayhole attacks. This supports the effectiveness of the SMOTE-Tomek technique in both balancing the dataset and reducing misclassification for underrepresented attack classes [5]. Furthermore, low misclassification rates across all categories confirm the model’s strong discrimination capability reducing both false positives and false negatives. This is particularly crucial in cybersecurity applications where false positives can waste resources, and false negatives can allow security breaches to go undetected.

The SMOTE–Tomek resampling approach also ensured consistent sensitivity to both frequent and infrequent attack classes, leading to a balanced intrusion detection system with strong precision and recall performance across all categories [3]. While the model achieved perfect scores (1.00) for Flooding attacks, this outcome requires cautious interpretation. It may result from highly distinguishable patterns within the Flooding class or potential overfitting influenced by synthetic samples generated during the SMOTE process. To avoid overconfidence, future studies should validate performance on external datasets and more varied attack scenarios to assess the robustness of Flooding attack detection under real-world variability.

While the evaluation results indicate clear performance improvements for the FA-DNN model, this study did not conduct formal statistical significance tests such as paired t-tests or confidence interval analysis to rigorously validate whether the observed differences are statistically meaningful. This decision was motivated by the computational demands of the Firefly Algorithm optimization, which required evaluating numerous hyperparameter configurations via repeated training. Nevertheless, the consistent improvements across multiple metrics (accuracy, loss, precision, recall, and F1-score) provide strong preliminary evidence of the model's effectiveness. Future research will aim to include statistical testing protocols to better quantify result robustness and confirm the generalizability of findings across different data splits. While the evaluation results indicate consistent performance improvements for the FA-DNN model, several limitations should be acknowledged.

First, the use of SMOTE-Tomek for class balancing introduces synthetic samples into the dataset, which may not fully reflect the variability of real-world network traffic. This could lead to overly simplified decision boundaries, particularly for attack types with low original prevalence. Second, the experimental evaluation was conducted solely on the WSN-DS dataset. Although it is a widely used benchmark, reliance on a single dataset limits the generalizability of the results. Testing the model on additional real-world or heterogeneous datasets would provide a more comprehensive validation. Third, formal statistical significance tests (e.g., confidence intervals, hypothesis testing) were not applied to the reported metrics due to the computational overhead associated with retraining multiple configurations during FA optimization. As a result, the reported performance gains should be interpreted as indicative rather than conclusive. Future research should aim to incorporate statistical validation and multi-dataset evaluation to further assess the robustness and reliability of the proposed approach.

## 7. Conclusions and Future Work

The contributions of this study are summarized briefly in this section in terms of key findings, and possible ways for expanding the cyberattack detection models will be suggested in the future research. The FA optimized deep learning model would significantly improve the cyberattack detection accuracy (97.82%) over traditional models. Effectively addressing class imbalance, a minority-class attack was represented equally by the SMOTE-Tomek technique. With our loss function, the model generalized better (i.e. fewer false positives or false negatives) by eliminating noisy samples and refining data distribution. At Epochs 4 and 9, the learning rate was adjusted dynamically by ReduceLROnPlateau mechanism to prevent weight updates with unnecessary and give stable convergence, which improves performance of the model.

Unlike conventional models, the optimized model did not suffer from performance degradation on training and validation datasets as normally observed, rather, it maintained a constant stable accuracy. Finally, the optimized model performed the best in terms of loss and accuracy compared to the existing machine learning models such as XGBoost, CNN, and SVM. Across stability and accuracy, the proposed FA-DNN model far surpassed XGBoost, CNN, SVM, and the rest of traditional models. An FA dynamically explores the search space, dynamically adapting hyperparameters based on performance trends in order to maximize a classification accuracy at minimal computational cost. As a result of its scalability and robust performance, the FA-DNN model is a very strong deep learning technique for real time cyber attack detection with outstanding classification accuracy, model stability and robustness from class imbalance.

While the Firefly Algorithm has proven effective for hyperparameter tuning, future research should explore Bayesian Optimization as a potential alternative. Bayesian Optimization, known for its probabilistic model-based approach, could provide a more sample-efficient and computationally cost-effective strategy. A comparative study between FA and Bayesian Optimization would offer deeper insights into their respective advantages in optimizing deep learning models for cybersecurity applications. as well as, to validate the practical effectiveness of the proposed model, future work should focus on deploying and evaluating it in real-world cybersecurity environments, including IoT security frameworks, industrial networks, and cloud-based infrastructures. Key evaluation metrics should include inference speed, false positive rates, adaptability to zero-day attacks, and computational efficiency in high-traffic conditions.

## 8. Declarations

### 8.1. Author Contributions

Conceptualization: N.A.H. and O.N.J.; Methodology: N.A.H.; Software: N.A.H.; Validation: N.A.H. and O.N.J.; Formal Analysis: N.A.H. and O.N.J.; Investigation: N.A.H.; Resources: O.N.J.; Data Curation: O.N.J.; Writing

Original Draft Preparation: N.A.H. and O.N.J.; Writing Review and Editing: N.A.H. and O.N.J.; Visualization: N.A.H.; All authors have read and agreed to the published version of the manuscript.

## 8.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

## 8.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

## 8.4. Institutional Review Board Statement

Not applicable.

## 8.5. Informed Consent Statement

Not applicable.

## 8.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] S. Abbas, I. Bouazzi, S. Ojo, A. Al Hejaili, G. A. Sampedro, A. Almadhor, and M. Gregus, "Evaluating deep learning variants for cyber-attacks detection and multi-class classification in IoT networks," *PeerJ Computer Science*, vol. 10, no. 1, pp. 1-12, 16 Jan. 2024, doi: 10.7717/peerj‑cs.1793.

[2] M. H. Behiry and M. Aly, “Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with AI and machine learning methods,” *Journal of Big Data*, vol. 11, article 16, no. 1, pp. 1–39, Jan. 2024, doi: 10.1186/s40537-023-00870-w.

[3] R. Alguliyev and R. Shikhaliyev, “Network cybersecurity incidents multiclassification based on deep learning,” *Informasiya texnologiyaları problemləri*, vol. 15, no. 2, pp. 16–23, Jun. 2024, doi: 10.25045/jpit.v15.i2.03

[4] T. Wongvorachan, S. He, and O. Bulut, “A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining,” *Information*, vol. 14, no. 1, p. 54, Jan. 2023, doi: 10.3390/info14010054.

[5] A.-j. Li and P. Zhang, "Research on Unbalanced Data Processing Algorithm Base Tomeklinks-Smote," in *Proc. 2020 3rd Int. Conf. Artificial Intelligence and Pattern Recognition (AIPR)*, New York, NY, USA: Association for Computing Machinery, vol. 2020, no. 1, pp. 13–17, 2020, doi: 10.1145/3430199.3430222.

[6] E. F. Swana, W. Doorsamy, and P. Bokoro, “Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset,” *Sensors*, vol. 22, no. 9, pp. 32-46, Apr. 2022, doi: 10.3390/s22093246.

[7] L. Chen and J. Li, “A Firefly Algorithm Based on Prediction and Hybrid Samples Learning,” in *Proceedings of the 2023 International Conference on Intelligent Computing (ICIC 2023)*, *Lecture Notes in Computer Science*, vol. 14086, Springer, Singapore, no. 1, pp. 262–274, Jul. 30 2023, doi: 10.1007/978-981-99-4755-3_23.

[8] X.-S. Yang and X. He, “Firefly algorithm: recent advances and applications,” *International Journal of Swarm Intelligence*, vol. 1, no. 1, pp. 36–50, Aug. 2013, doi: 10.1504/IJSI.2013.055801.

[9] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level Hybrid Support Vector Machine and Extreme Learning Machine Based on Modified K-means for Intrusion Detection System," Expert Systems with Applications, vol. 67, no. 1, pp. 296–303, Jan. 2017, doi: 10.1016/j.eswa.2016.09.041.

[10] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, “A survey of deep learning methods for cyber security,” *Information*, vol. 10, no. 4, article 122, 2 Apr. 2019, doi: 10.3390/info10040122.

[11] Z. H. E. Wang, C. Wu, K. Zheng, X. Niu, and X. Wang, “SMOTE-Tomek-Based Resampling for Personality Recognition,” *IEEE Access,* vol. 7, no. 9, pp. 129678–129689, Sep. 2019, doi: 10.1109/ACCESS.2019.2940061

[12] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," *IEEE Access*, vol. 9, no. 1, pp. 114381-114391, 2021, doi: 10.1109/ACCESS.2021.3104357.

[13] G. M. Dandime, R. Deshmukh, S. Patil, S. R. Waghmare, and A. R. Raut, "Energy enhancement and optimization of WSN using firefly algorithm and deep learning," *in Proc. 2022 Int. Conf. Edge Comput. Appl. (ICECAA), IEEE,* vol. 2022, no. 1, pp. 1432–1436, 2022, doi: 10.1109/ICECAA55415.2022.9936146.

[14] M. Alsoul, H. Zaher, N. Ragaa, and E. M. Oun, "A new efficient hybrid approach for machine learning-based firefly optimization," *Iraqi Journal of Science,* vol. 64, no. 9, pp. 4600–4612, 2023, doi: 10.24996/ijs.2023.64.9.24

[15] O. A. Beg, A. A. Khan, W. U. Rehman, and A. Hassan, "A review of AI-based cyber-attack detection and mitigation in microgrids," *Energies,* vol. 16, no. 22, p. 7644, 2023, doi: 10.3390/en16227644.

[16] B. J. Kim, H. Choi, H. Jang, D. Lee, and S. W. Kim, "How to use dropout correctly on residual networks with batch normalization," *in Proceedings of the 40th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR*, vol. 216, no. 1, pp. 1583–1592, 2023.

[17] K. Medvedieva, T. Tosi, E. Barbierato, and A. Gatti, "Balancing the Scale: Data Augmentation Techniques for Improved Supervised Learning in Cyberattack Detection," *Eng.*, vol. 5, no. 3, pp. 2170–2205, 2024, doi: 10.3390/eng5030114.

[18] L. Huang, "Motivation and overview of normalization in DNNs," in *Normalization Techniques in Deep Learning, Springer*, vol. 2022, no. 1, pp. 11–18, 2022, doi: 10.1007/978-3-031-14595-7_2.

[19] X. Qi, Y. Wei, X. Mei, R. Chellali, and S. Yang, "Comparative analysis of the linear regions in ReLU and LeakyReLU networks," *in Neural Information Processing (Communications in Computer and Information Science, vol. 1962), B. Luo et al., Eds., Singapore: Springer Nature*, vol. 2024, no. 1, pp. 528–539, 2024, doi: 10.1007/978-981-99-8132-8_40.

[20] F. Rahma, M. C. Rajasa, R. F. Rachmadi, B. A. Pratomo, and M. H. Purnomo, "Resampling Effects on Imbalanced Data in Network Intrusion Classification," *in Proc. 2024 Int. Electronics Symp. (IES), IEEE,* vol. 2024, no. 1, pp. 534–540, 2024, doi: 10.1109/IES63037.2024.10665861.

[21] H. M. Saleh, H. Marouane, and A. Fakhfakh, "Improves intrusion detection performance in wireless sensor networks through machine learning, enhanced by an accelerated deep learning model with advanced feature selection," *Iraqi J. Comput. Sci. Math.*, vol. 5, no. 3, pp. 1-23, 2024.

[22] D. Kilichev and W. Kim, "Hyperparameter Optimization for 1D-CNN-Based Network Intrusion Detection Using GA and PSO," *Mathematics,* vol. 11, no. 17, article 3724, pp. 1-12, 2023, doi: 10.3390/math11173724.

[23] A. Kumar, D. K. Gupta, S. R. Ghatak, and S. R. Prusty, "A Comparison of PSO, GA and FA-Based PID Controller for Load Frequency Control of Two-Area Hybrid Power System," *in Smart Technologies for Power and Green Energy, R. N. Dash, A. K. Rathore, V. Khadkikar, R. Patel, and M. Debnath, Eds., Lecture Notes in Networks and Systems*, vol. 443, Singapore: Springer, vol. 2023, no. pp. 321–334, 2023, doi: 10.1007/978-981-19-2764-5_23.

[24] K. Jain, R. P. Yadav, and M. Raza, "Understanding High-Dimensional Bayesian Optimization," *arXiv preprint* arXiv:2502.09198, vol. 2024, no. 1, pp. 1–17, Feb. 2024, doi: 10.48550/arXiv.2502.09198.