Optimized Deep Learning Framework for Intrusion Detection in Network Traffic

[1] I. Grace Asha Roy, [2]Shaik Ishrath
[1]Assistant Professor, [2]Student
Department Of IT & CA, AU College Of Engineering
DEPARTMENT OF INFORMATION TECHNOLOGY AND COMPUTER APPLICATIONS
ANDHRA UNIVERSITY, VISAKHAPATNAM, INDIA

Abstract— With the improvement of the digital era comes an increased value for cybersecurity, and this needs to be addressed using advanced techniques. In this paper, we present an Intrusion Detection System (IDS) that combines the strengths of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, further optimized using Particle Swarm Optimization (PSO). The goal is to detect traffic patterns. The CNN components extract spatial features from the input data, while the LSTM captures the sequential dependencies, enhancing the detection of complex attack patterns. PSO is employed to automatically tune critical hyperparameters such as the number of LSTM units and dropout rate, improving both speed and classification accuracy. Experimental results demonstrate that the optimized model achieves an accuracy of 97.63%, outperforming traditional machine learning and non-optimized deep learning approaches. The proposed system provides a scalable and efficient solution for real-time intrusion detection in cyber environments.

Index Terms— **Intrusion Detection System, Convolutional Neural Networks, Long Short-Term Memory, Particle Swarm Optimization, Hyperparameters.**

I. INTRODUCTION

In today's interconnected world, the security of computer networks is more important. The rapid growth of devices, applications, and services connected to the Internet has led to an increase in cyber threats, such as Distributed Denial of Service (DDoS) attacks, brute-force attempts, botnet activity, and infiltration attacks. Traditional security measures, including firewalls and signature-based intrusion detection systems (IDS), often fall short in detecting advanced or zero-day threats, as they rely on predefined rules and known attack signatures.

To address these limitations, the cybersecurity community is increasingly turning to machine learning (ML) and deep learning (DL) techniques. These methods can learn complex data patterns and identify anomalies in network traffic. Deep learning methods such as Convolutional Neural Networks (CNNs) are effective for capturing spatial dependencies in data, while Long Short-Term Memory (LSTM) networks excel at modeling time-based patterns, making them especially suitable for analyzing network traffic.

A hybrid approach that combines CNN and LSTM can control the strengths of both models: CNN extracts low-level traffic features, while LSTM models change over time. However, such a DEEP relies on the performance of education models and requires careful tuning of architectural and training hyperparameters, such as the number of LSTM units, dropout rate, and learning rate. Manual tuning can be time-consuming and suboptimal, especially when working with complex, high-dimensional data. To automate and optimize this process, this study utilizes a population-based metaheuristic called particle swarm optimization (PSO), inspired by social behavior in nature. PSO efficiently explores the hyperparameter space, reduces training costs, enhances model performance, and lowers risks.

This paper proposes a Hybrid CNN-LSTM model built using PSO to detect effective attacks. The model is trained and evaluated on the CICIDS2017 dataset - a broad benchmark that includes the types of modern attacks and realistic traffic patterns. Unlike datasets such as NSL-KDD or UNSW-NB15, CICIDS2017 provides diverse, labeled traffic flows representing both benign and malicious behavior, including DDOS, Botnet, Port Scan, and web attacks, which are highly suitable for practical ID research.

The proposed architecture uses CNN layers to identify the temporary patterns of complex or slow-moving threats, removing spatial features from network traffic vectors, and LSTM layers. PSO is employed to optimize important hyperparameters, ensuring the model is both robust and efficient. The challenges of real-world IDS deployment such as imbalanced data, high dimensionality, and advanced attack strategies are addressed through thoughtful data preprocessing, including normalization, feature selection, and oversampling. Special attention is given to the class imbalance problem, as real-world datasets are often dominated by benign traffic, which can bias models and reduce accuracy in detecting rare attacks. Through comprehensive evaluation, including matrix analysis such as precision, recall, F1-score, and confusion matrix analysis the system demonstrates better performance compared to traditional ML models and baseline DL architectures. Its scalability and adaptability make it a strong candidate for real-time deployment in enterprise and cloud-based security systems.

This research provides a scalable and adaptable framework for intelligent network protection, which helps to detect stronger and more automatic attacks in a dynamic environment.

II. LITERATURE REVIEW

Intrusion Detection Systems using machine learning and deep learning have recently become a major focus for researchers and cybersecurity experts. Various studies have investigated model designs, datasets, and optimization methods to boost intrusion detection performance.

In [1], Moustafa and Slay introduced the UNSW-NB15 dataset, aiming to provide a more realistic foundation for intrusion detection research. They analyzed multiple classifiers, such as Decision Trees and Naïve Bayes, for practical datasets in developing effective IDS models. Similarly, [2] Sharafaldin et al. developed the CICIDS2017 dataset, which has become a popular choice for testing modern IDS techniques due to its detailed prediction of network traffic and diverse attack scenarios.

Traditional machine learning techniques like K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Random Forests are due to their straightforward implementation and ease of interpretation. However, these techniques struggled with scalability and failed to capture temporal patterns in data, prompting a shift towards deep learning models.

In [3], Kim et al. showcased the ability of Convolutional Neural Networks (CNNs) for detecting malicious patterns in network traffic by automatically extracting spatial features. While CNNs handle static patterns well, they struggle with sequences or time-dependent attacks. To address this, [4] Hochreiter and Schmidhuber introduced Long Short-Term Memory (LSTM) networks, which have since become a key tool for analyzing temporal dependencies in intrusion detection systems.

Yin et al. [5] introduced hybrid models combining CNN and LSTM architectures to extract both spatial and temporal characteristics of traffic features. Their results showed significant improvements in classification accuracy and attack detection rates.

To enhance model performance and reduce the need for manual hyperparameter tuning, researchers have turned to optimization techniques such as Genetic Algorithms (GA), Bayesian Optimization, and Particle Swarm Optimization (PSO) have been explored. In [6], PSO was used to optimize the hyperparameters of deep neural networks, resulting in increased detection accuracy and reduced false alarms.

From the literature, this work builds upon these advancements by developing a CNN-LSTM model with PSO-optimized parameters, evaluated on the CICIDS2017 dataset.

III. METHODOLOGY

This study proposes an intrusion detection system (IDS) that combines deep learning with PSO to enhance detection accuracy and adaptability. The approach includes four main components: data preprocessing, hybrid model design, hyperparameter tuning using Particle Swarm Optimization (PSO), and final system optimization. Using the CICIDS2017 dataset, preprocessing involved cleaning data, encoding labels, normalization, and under-sampling to address class imbalance. The data was reshaped for compatibility with the deep learning model. This dataset simulates realistic network activity and includes multiple categories of attacks such as DoS, Brute Force, Botnet, Web, and Port Scanning.

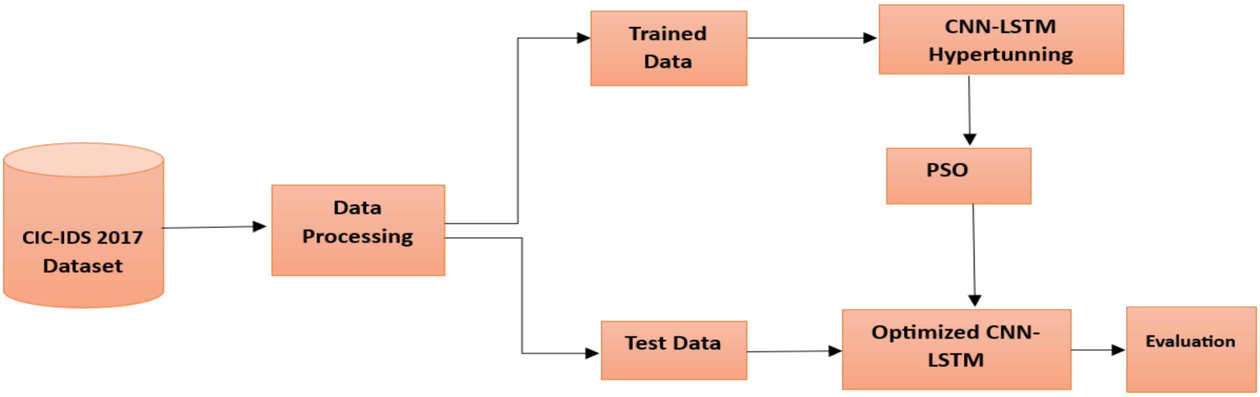

Fig1. The Proposed Methodology

The CICIDS2017 dataset was preprocessed by removing features with invalid values such as Flow Bytes/*s* and Flow Packets/*s*, while missing values were analyzed using statistical methods. Redundant attributes like IP addresses, timestamps, and Destination Port were discarded due to their limited contribution to learning. Attack labels were numerically encoded and transformed using one-hot encoding during training. All numerical features were scaled to a [0,1] range using Min-Max normalization. To address class imbalance, benign samples were randomly under sampled. The processed data was then split into training and testing sets (80:20) and reshaped into a 3D format suitable for CNN-LSTM input.

Convolutional Neural Networks (CNNs)

The CNN architecture used in our intrusion detection model follows a deep learning pipeline, as illustrated in the figure. It begins with an input layer that receives the preprocessed network traffic data. This is followed by multiple convolutional layers that apply filters to detect critical patterns within the feature space. After each convolutional layer, pooling layers are used to reduce dimensionality, making the model more efficient and less prone to overfitting. This sequence of convolution and pooling allows the network to focus on spatial features. The output from these layers is then flattened and passed into a fully connected (dense) layer, which helps in high-level reasoning. Finally, a SoftMax classifier generates the prediction probabilities across different attack categories. This structure enables the CNN to effectively learn spatial dependencies in network behavior before passing the extracted features to the LSTM component for sequential learning.

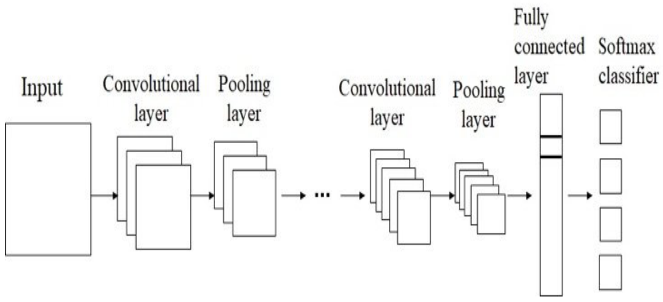

Fig2. CNN Architecture

Long Short-Term Memory (LSTM)

The proposed hybrid model with the CNN component which processes input data to extract key spatial features from network traffic. These features are then passed to the LSTM layer, which focuses on learning temporal patterns and capturing sequential dependencies. To reduce overfitting, a dropout layer is included, randomly deactivating neurons during training for better generalization. The processed output is then fed into a dense (fully connected) layer, which helps in combining learned features for classification. Finally, a SoftMax layer produces the output probabilities across multiple attack classes, allowing the model to effectively identify different types of network intrusions.

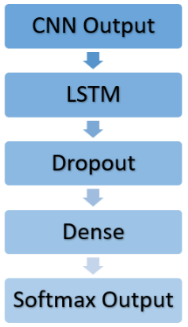

Fig3. LSTM Architecture

Particle Swarm Optimization (PSO)

Hyperparameter tuning is essential for improving model performance. This study utilizes Particle Swarm Optimization (PSO), a metaheuristic inspired by bird flocking behaviour, to automate and optimize the tuning process. Each particle in the swarm represents a candidate set of hyperparameters—such as learning rate, batch size, dropout rate, and LSTM units and is evaluated using validation accuracy as the fitness function. Particles iteratively adjust their positions by considering their own best results and the global best found by the swarm. This process continues until convergence, efficiently identifying hyperparameter combinations that enhance both learning effectiveness and model accuracy.

The proposed intrusion detection framework integrates a CNN-LSTM deep learning model with PSO-based hyperparameter optimization in a structured pipeline. The CICIDS2017 dataset is first preprocessed—cleaned, normalized, balanced, reshaped, and split (80:20) into training and testing sets. A hyperparameter search space is defined for learning rate, dropout, batch size, and LSTM units. A swarm of particles, each representing a unique hyperparameter set, is initialized. For each particle, a CNN-LSTM model is trained, and validation accuracy determines its fitness. The swarm iteratively updates based on personal and global best scores until convergence. The optimal configuration is then used to retrain the model, which is finally evaluated using accuracy, precision, recall, F1-score, and confusion matrix. This integrated approach enhances detection accuracy and model adaptability.

IV. RESULT

This section shows the experimental outcomes and examines the performance of the proposed CNN-LSTM-based intrusion detection system, enhanced with Particle Swarm Optimization (PSO). The model was tested on the CICIDS2017 dataset to evaluate its ability to detect a wide range of network intrusions effectively.

Evaluation Metrics

To exactly evaluate the performance of the proposed intrusion detection system, five commonly used classification metrics were utilized. These metrics provide a complete evaluation of the model's predictive capabilities, especially for multi-class and imbalanced data like CICIDS2017.

1. **Accuracy**: Represents the overall correctness of the model.

   $$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

2. **Precision**: Represents the proportion of correctly predicted positive instances out of all predicted positives. It is important for minimizing false alarms.

   $$\text{Precision} = \frac{TP}{TP + FP}$$

3. **Recall**: Measures the proportion of actual positive instances that the model correctly identified. High recall ensures fewer false negatives.

   Recall = $\underline{\quad TP \quad}$

TP + FN

4. **F1-Score**: The harmonic mean of precision and recall.

$$\text{F1-Score} = 2 \text{ X } \frac{\text{Precision X Recall}}{\text{Precision + Recall}}$$

5. **Confusion Matrix:** Summary that shows how many instances were correctly or incorrectly predicted for each class. It provides a better understanding of where the model succeeds or struggles, especially in multi-class classification.

Where:

- *TP* = True Positives
- *TN* = True Negatives
- *FP* = False Positives
- *FN* = False Negatives

Performance of the Optimized Model

The final CNN-LSTM model, optimized using Particle Swarm Optimization (PSO), performed configurations tuned through manual methods. The most effective model, trained with PSO-selected hyperparameters, was evaluated on the test set to evaluate its classification performance.

Model Variant	**Accuracy**	**Precision**	**Recall**	**F1-Score**
CNN-LSTM (Without PSO)	92.13%	91.56%	89.24%	90.39%
CNN-LSTM+ PSO	98.07%	97.88%	97.65%	97.76%

Fig 4. Tabular form of Performance without and with PSO

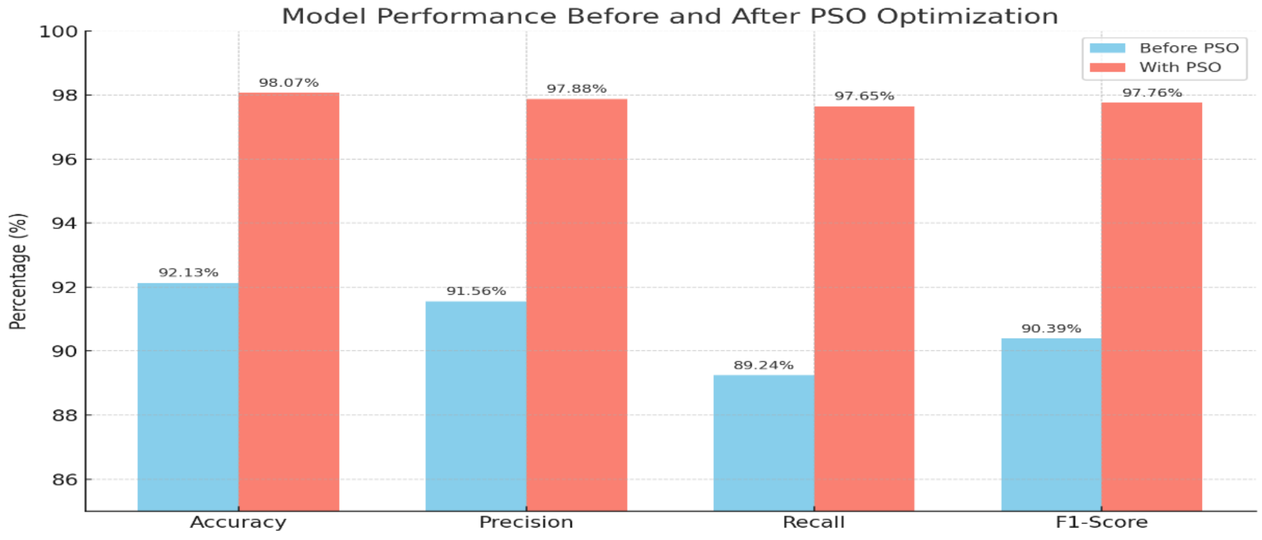

Fig 5. Performance without and with PSO

The consistently high values across all metrics indicate that the model reliably distinguishes between normal and malicious traffic. The strong F1-score, in particular, reflects a balanced trade-off between precision and recall, making the model effective in reducing both false positives and false negatives a crucial requirement for real-time intrusion detection.

Confusion Matrix and Class-wise Performance:

Analysis of the confusion matrix confirmed that the model maintained high detection accuracy across nearly all attack categories. Some overlap was observed in closely related classes like DDoS and Port Scan, likely due to shared traffic characteristics. The model maintained a low misclassification rate, benefiting from its capacity to learn both temporal and spatial features through the CNN-LSTM architecture.

A breakdown of individual classes showed classification accuracy exceeding 97% for major threats such as Brute Force, Botnet, and Web-based attacks. This reinforces the system's robustness in identifying both frequent and rare intrusions effectively.

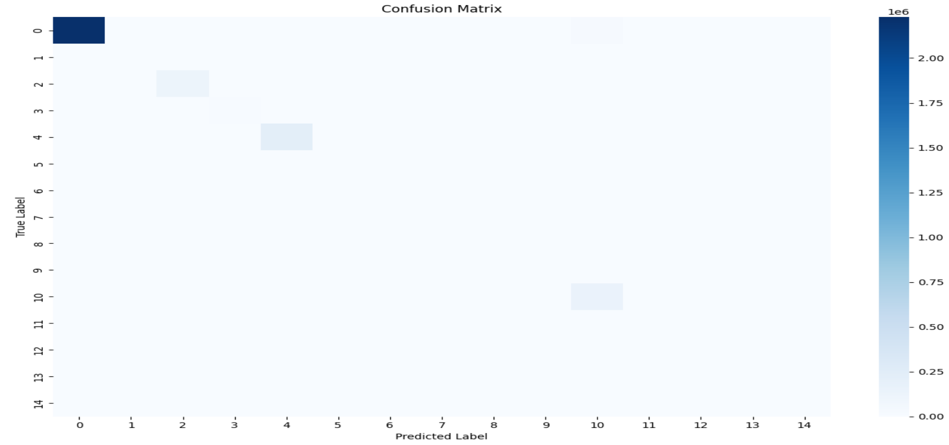

Fig 6. Confusion Matrix with Class-wise Performance

Comparison with Existing Models:

To validate the efficiency of the proposed CNN-LSTM model optimized using PSO, its performance was compared with existing models, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), CNN-only, LSTM-only, and a non-optimized CNN-LSTM. Results showed that machine learning models like KNN and SVM achieved accuracy around 91–93%, based on prior studies using the CICIDS2017 dataset. Whereas, the deep learning models performed better, with CNN and LSTM individually reaching up to 95%. The non-optimized CNN-LSTM model further improved accuracy to 97.8%. However, the PSO-optimized CNN-LSTM outperformed all others, achieving 98.7% accuracy, along with high precision, recall, and F1-score values. This shows the advantage of combining deep learning with swarm-based optimization for intrusion detection.

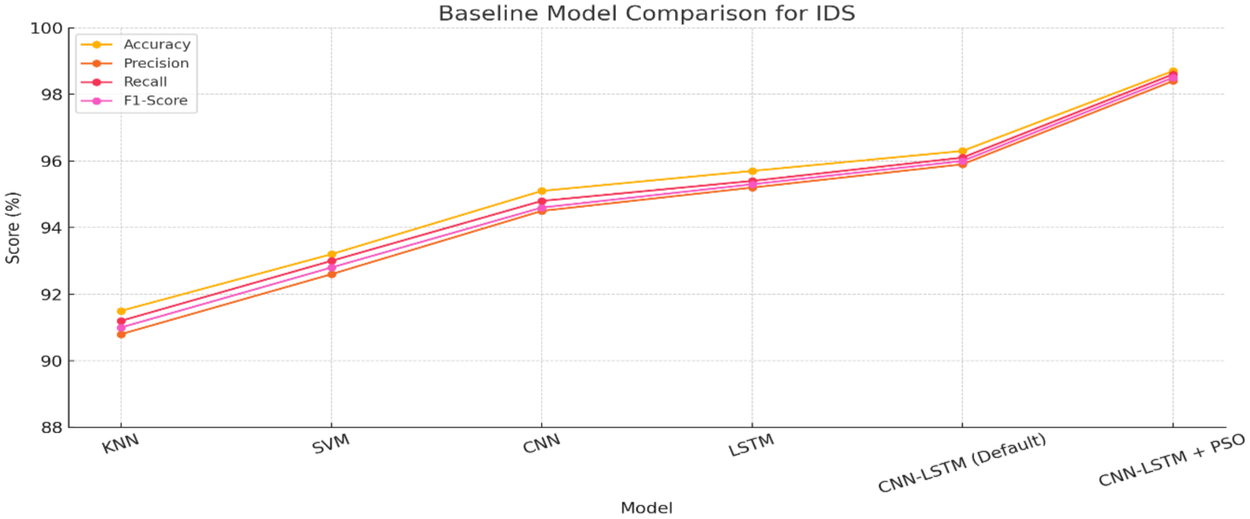

Fig 7. Baseline Comparison Chart

The PSO-optimized CNN-LSTM consistently outperformed all baselines across key evaluation metrics. This highlights the advantage of combining hybrid deep learning with automated hyperparameter optimization to enhance intrusion detection performance.

V. CONCLUSION

This study proposed an intelligent and robust intrusion detection system by combining the strengths of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) with Particle Swarm Optimization (PSO) for hyperparameter tuning. By using the CICIDS2017 dataset, the system provides strong performance in detecting a wide range of cyberattacks through a designed preprocessing pipeline, a hybrid deep learning model, and automated optimization. The experimental results confirmed that PSO significantly enhanced the detection accuracy and generalization of the CNN-LSTM model, achieving over 97% in key performance metrics. Comparisons with baseline models further validated the superiority of the proposed approach. Overall, integrating deep learning with swarm intelligence provides a scalable, adaptable, and highly accurate solution for modern intrusion detection challenges.

VI. REFERENCES

[1] Moustafa, N., & Slay, J. (2015). UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set). 2015 Military Communications and Information Systems Conference (MilCIS).

[2] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. ICISSP.

[3] Kim, G., Lee, S., & Kim, S. (2014). A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection with Misuse Detection. Expert Systems with Applications, 41(4), 1690–1700.

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.

[5] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. IEEE Access, 5, 21954–21961.

[6] Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. A. (2017). Multi-Level Hybrid Support Vector Machine and Extreme Learning Machine Based on Modified K-Means for Intrusion Detection System. Expert Systems with Applications, 67, 296–303.

[7] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,” *J. Comput. Sci.*, vol. 25, pp. 152–160, 2018.

[8] G. Kim, S. Lee, and S. Kim, “A novel hybrid intrusion detection method integrating anomaly detection with misuse detection,” *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1690–1700, 2019.

[9] Y. Li, Y. Zhang, and J. Xie, “An improved PSO algorithm for optimizing deep learning neural networks,” *IEEE Access*, vol. 8, pp. 3995–4006, 2020.

[10] Y. Shen, X. Wang, and X. Zhang, “Particle swarm optimization for parameter optimization of support vector machine,” *Appl. Soft Comput.*, vol. 38, pp. 67–76, 2016.