ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL
JALT

SWARM INTELLIGENCE FOR REAL-TIME INTRUSION DETECTION:A BIO-INSPIRED FRAMEWORK USING ANT AND BEE COLONY OPTIMIZATION"

Ahmed Sohaib Khawer
Network/System Engineer,
Punjab Information Technology Board (PITB) –
Lahore, Punjab
sohaib.khawer@gmail.com

Abstract:
As these threats become more advanced and new ones appear, relying on constant threat detection is more important for network security. Traditional IDS often have problems adapting and reacting quickly to large and diverse environments. This document discusses a bio-inspired design for IDS that uses Ant Colony Optimization (ACO) and Bee Colony Optimization (BCO) algorithms to perform real-time threat scanning. Focusing on how natural swarms are decentralized and self-organized, the system models network traffic analysis as a combined effort to detect anomalies while locating sources. A new framework is built that eases the routing of agents, increases the importance of selected features and improves detection accuracy while using few resources. Tests done using the NSL-KDD and CICIDS2017 datasets have shown that the bio-inspired IDS achieves high accuracy, few false alarms and better ability to adapt than classical machine learning models. We have found that using swarm intelligence is a suitable and scalable way for building better IDS, fit for protecting systems in modern, active cybersecurity settings.

Keywords: Swarm Intelligence, Intrusion Detection System, Ant Colony Optimization, Bee Colony Algorithm, Real-Time Threat Detection, Bio-Inspired Computing

1. Introduction

1.1 Importance of Real-Time Threat Detection in Modern Networks

Since digital communication and infrastructure are always changing, real-time threat detection is now very important. Today, networks in finance, healthcare and national security often face hackers who use malicious software, phishing methods, zero-day vulnerabilities and Advanced Persistent Threats (APTs). Because these systems have grown so advanced, prompt detection and action against attacks is extremely important. The previous approach to studying incidents after they happen cannot meet today's changing needs, due to the rising cost of breaches. And so, intrusion detection systems (IDS) should move towards being fast, flexible and operating independently.

1.2 Limitations of Traditional and Signature-Based IDS

Most of the traditional IDS systems belong to either the signature-based or anomaly-based categories. Signature-based IDS such as Snort depend on having patterns that match known attack signatures. They work well at finding threats that are already listed, but they cannot handle unique or hidden threats and must be updated regularly. This type of IDS relies on considering what is normal in activities, so it can detect new types of threats. Even so, this system regularly results in high false positives and takes a lot of computer power to reflect complex behaviors in real time. Both methods usually use centralized detection models which may lead to reduced performance or increased risk of failure for the whole system.

1.3 Emergence of Bio-Inspired Computing in Cybersecurity

To address the limitations of standard systems, researchers have looked to bio-inspired computing which imitates traits from nature to address tough computational tasks. As they are based on evolution, immune systems and animal societies, bio-inspired algorithms ensure cybersecurity by allowing adaptation, robustness and sharing responsibility without a central system. They do not need to use specific signatures or set models for their analysis. They can believe, behave and function just like real organisms in different and volatile situations. This

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL
JALT

approach has given rise to swarm intelligence which helps create proactive IDS that can find threats on their own.

1.4 Swarm Intelligence: Natural Principles and Computational Relevance

Swarm intelligence is the way many independent and self-organizing systems work together. Examples in nature are ant colonies finding the best way to food, birds flying together in a group and bees hunting for nectar. Simple agents that work according to some basic rules and only interact in their surroundings can solve big challenges experienced everywhere. Swarm intelligence is used in computing to handle optimization, routing, scheduling and pattern recognition. Ant Colony Optimization (ACO) and Bee Colony Optimization (BCO) are examples of algorithms modeled after natural swarming behaviors. ACO uses pheromone trails to assist agents in finding their way, so it is particularly suited for finding routes and paths. The model describes bee behavior in food collection, giving them a good way to alternate between looking for new places and focusing on one area. With these features, swarm algorithms can work efficiently and are suitable for using in IDS systems in real time and distributed environments.

2. Related Work

2.1 Traditional Intrusion Detection Approaches

Most Intrusion Detection Systems (IDS) have used two approaches—signature-based and statistical—to identify harmful actions within network traffic. Some systems like Snort and Bro (now Zeek) inspect network traffic against a database containing known malicious signatures. Even though they detect viruses perfectly, these systems cannot prevent zero-day or constantly evolving malware. Different from those methods, statistical approaches use usual traffic behaviors as a model and point out any abnormal activities as possible intrusions. While more flexible, they sometimes lead to a lot of errors and are hard to keep up with the changes in current networks. Both ways of defending cyber security struggle when it comes to handling new attack methods fast and on a large scale.

2.2 Overview of Swarm Intelligence

Swarm intelligence (SI) is the way systems made up of individuals, without leadership, work together in groups inspired by ants, bees, birds and fish. Because of this way information is passed among individuals in natural systems, they serve as excellent examples for studying distributed problem-solving in computers. Some of the main algorithms in SI used in cybersecurity are called Ant Colony Optimization (ACO), Bee Colony Optimization (BCO) and Particle Swarm Optimization (PSO). Each algorithm matches certain actions from animals: ACO represents a trail of pheromones left by ants and how they search for food; BCO takes its ideas from bees, including scouting and the way they use dance signals; PSO copies the minor and major adjustments particles (agents) in a system make in response to their knowledge and experience. Their ability to adjust, handle increase in size and stay stable makes them perfect for bettering existing IDS.

2.3 Previous Applications of Swarm Intelligence in IDS

Almost all of the work with swarm intelligence in IDS is done for things like choosing what features to use, setting up parameters and spotting anomalies. ACO extracts important features from large datasets which has enhanced the accuracy of classifying in NSL-KDD and CICIDS2017. Joining ACO and SVM has helped improve accuracy and decreased how much time and space are needed in calculations. BCO plays a role in detecting anomalies and categorizing them, frequently showing greater flexibility and ability to work well compared to standard approaches. Because of its quick speed and simple use, PSO is often applied to set the thresholds or adjust the parameters of an Intrusion Detection System. Merging swarm intelligence and either fuzzy logic or neural networks allows some approaches to detect a

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL
JALT

higher number of threats. Nonetheless such systems are mostly kept offline and will not adjust to new or changing real-time data.

2.4 Comparative Analysis of Bio-Inspired Methods in Cybersecurity

Extensive studies show that bio-inspired methods, especially swarm-based algorithms, are regularly compared to traditional ML and deep learning. While CNN and LSTM models are really powerful, they consume a lot of computing power and are hard to understand which is the reason they should not be used in areas where processing speed or resources are low. In contrast, swarm intelligence gives computers systems that are both light, clear to understand and work in parallel. It has been found in comparative research that ACO is stronger at selecting which features to use, while PPO does better at adjusting hyperparameters. BCO has been able to learn from data in noisy environments. However, difficulties arise for them when we look at real-time performance, how well they connect with security tools and how flexible they are in multiple service areas. It is difficult to make these evaluation methods consistent and the findings change a lot because of differences in datasets, classifiers and the metrics used.

2.5 Research Gaps

Even with good results, swarm intelligence has not been fully applied in IDS systems. Existing approaches mostly concentrate on finding the best solution offline, not on identifying it as soon as it happens. Very rarely are swarm models put directly into the decision mechanism of an IDS, so they cannot respond to novel threats at once. There are also very little studies investigating how the various kinds of agent interactions in ACO and BCO, for example, could guide the structure of networked systems or facilitate coordination in a distributed IDS. Known swarm-based IDS models have not been tested enough in real-world usage such as in IoT, multi-cloud or edge situations. The area of combining swarm-based detection with SIEM tools or automated responses still needs more study. This shows that it will be important to research how swarm intelligence can work as a real-time system, not just as an optimizer, in places with a lot of data flowing in at once.

3. Methodology

3.1 System Architecture Overview

The Intrusion Detection System (IDS) that I have proposed uses a modular design so that swarm intelligence can help it spot threats quickly and effectively. The first step in the system pipeline is to gather real network traffic and this is followed by normalizing, filtering noise and separating data based on its protocol type. After cleaning, features are looked for in the data that are specific to how attacks happen.

Next, Ant Colony Optimization (ACO) or Bee Colony Optimization (BCO) are applied to top security algorithms—to quickly detect the most important features and organize suspicious actions. It keeps working all the time, changing pheromone deposits (in ACO) or recruitment rates (in BCO) according to continuous input. The final step is to pull in swarm suggestions along with classifier assessments to signal threats or to update the blacklist. With its architecture, able to set up systems using the cloud or at edge locations, Zoom supports both large-scale use and ultra-fast response.

3.2 Dataset Description

NSL-KDD and CICIDS2017 are commonly used datasets that were used to check the system's effectiveness. It is more refined and balanced because it corrects redundant records and class imbalances from the original KDD'99 benchmark. It groups network behaviors and attacks into four groups called DoS, Probe, R2L and U2R.

Unlike CICIDS2012, the CICIDS2017 dataset contains recent data about DDoS, brute force, botnet and infiltration attacks. It includes useful data such as when each packet was sent, the amount of time each flow lasted and vital statistics which are excellent for examining network behavior in time series. Preprocessing the data meant removing blank fields, turning categorical

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL
JALT

features (like protocol type) into numbers for label encoding and standardizing all input attributes to make sure the model is stable. Also, a smaller real-time traffic simulation was done with Wireshark and tcpreplay to test the performance under live streaming, replicating web surfing, file downloads and simulated assaults.

3.3 Threat Modeling and Labeling Strategy

It is essential for the design of a real-time IDS to tell the difference between friendly and harmful behaviors on multiple scalable levels. In this study, traffic features identified are connected to particular tactics and techniques from the MITRE ATT&CK framework. Original classes from NSL-KDD and CICIDS2017 were used and the rest of the classes were marked manually using a system where one virtual machine generated attacks and the other received and logged them.

Using temporal correlation and session reconstruction, it was possible to link several packets or flows to just one kind of activity, for clear identification of threats. Having this rich schema helped the swarm models learn from changes in behavior which made them able to flag unusual behaviors instead of just warning about static patterns. Hence, if a computer keeps sending port scanning packets, it will be detected by swarm heuristics as a progressing threat rather than a single spike.

3.4 Swarm Algorithm Description

Researchers decided to use Ant Colony Optimization (ACO) since it is proficient at finding solutions and adjusting them based on feedback. ACO imitates ants in choosing the fastest route to food using the trails of pheromones they leave. In this case, the virtual ant means an approach made of a bunch of possible candidate features describing how traffic can be handled. While exploring the feature space, ants change the pheromone values depending on whether the current sample has the correct label or shows some anomaly

Δτijis the level of pheromone on the second feature after walking the first path, ρρ is the level that evaporates and ΔτijΔτij indicates what extra pheromone is deposited when the system detects accurately. Both the most scent-filled spots and the less investigated places are considered by using probability methods for feature selection.

Bee Colony Optimization (BCO) gets its ideas from how honey bees decide together. Various scout bees visit different locations and the employed and onlooker bees figure out which mix of features are best. To find out if bees' findings are accurate, the reward function uses precision, recall and F1-score. When a group is recruiting, they are all focused on finding the best features.

The function of IDS is to make rules through an algorithm over and over, so it is able to respond to changes and new threats quickly on the network. These approaches are better at handling data streams by making pheromone changes and shifting their way of recruiting.

Table 1: Summary of Swarm Intelligence Parameters Used in IDS

Algorithm	Key Parameters	Parameter Values	Role in Detection
ACO	Pheromone decay rate, α, β, number of ants	0.5, 1, 2, 50	Guides optimal path for packet behavior analysis
BCO	Recruitment probability, abandonment rate	0.7, 0.2	Controls bee communication & threat cluster detection

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL
JALT

PSO	Inertia weight, cognitive/social coefficients	0.6, 1.5/1.5	Adjusts agent velocity to converge on threat zones

3.5 Feature Extraction and Behavior Modeling

To identify useful features in the packet or flow data, statistical and timing-based analyses were done. Among them are inter-arrival times, how entropy affects the protocol, the range in packet sizes, how often the same IPs appear as source or destination and how long a connection lasts. Aggregating ants' or bees' behavior is done by following a rolling window approach to represent the same idea as a foraging ant or a bee charting their paths.

After feature vectors were created, we turned them into graphs, where each node stood for a group of related features (e.g., header fields of TCP, timing information) and edges represented how often those groups appeared together. Here, the swarm agents (representing ants or bees) used their actions to assess how helpful each feature was for spotting anomalies.

It also added two elements: when things happen in a day and how often they occur, to improve the swarm's memory. This feature helped the IDS identify successful and usual actions (e.g., backups) from odd ones that may suggest an attack is taking place (e.g., DDoS). With time, the swarm heuristic adapted itself by considering the number and severity of threats, forming a defensive mechanism that kept changing.

3.6 Real-Time Implementation Techniques

A stream processing architecture was used to add swarm models to the pipeline: streams were received using Apache Kafka, then Apache Flink or Spark Streaming was used for fast analysis of threats. Python, Scapy, PyShark, Dask and Numba were used to build the feature extraction modules and parallelize swarm computations, respectively.

Instead of focusing on each packet, the swarm models took traffic flows, grouped them, then processed those data groups. Changes in pheromones or fitness were made using EWMA to react more quickly. Also, if an anomaly was detected with enough confidence during the early stages, the search was stopped using early-exit criteria.

Reports from the IDS went to Grafana which security analysts used to view swarm tracking, the strength of the pheromone signals and any detected unusual activity. It lets analysts both take appropriate actions and learn from the system about how it reached its results.

4. Experimental Setup and Results

4.1 Tools, Programming Environment, and Hardware Setup

Open-source tools and the researchers' own created modules were used to bring the detection of intrusions to life in the framework. To perform data ingestion and preprocessing, Python 3.10 was used and Pandas, Scikit-learn and NumPy were helpful for feature engineering and statistical processing. PyShark (a Python version of tshark from Wireshark) gathered network packets and Apache Kafka made it possible to store them in real time.

From the outset, both Ant Colony Optimization (ACO) and Bee Colony Optimization (BCO) were designed to be flexible when managing pheromone levels and behaviors. Efficiency in fitness evaluations and search operations was boosted by using Numba (just-in-time interpretation) and running Dask on multiple threads at the same time.

Every test was carried out on a workstation equipped with an Intel Core i9-12900K CPU, 64GB DDR5 RAM and an NVIDIA RTX 3090 GPU. I installed it on Ubuntu 22.04 LTS which is built to work smoothly with the libraries needed for smooth processing. Because the data should be examined non-stop, the review was done with Apache Flink, since it is tolerant of errors and can track data.

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL
JALT

4.2 Baseline Models for Comparison

The efficacy of a bio-inspired IDS was tested by comparing it to numerous popular machine learning models. These included:

- Support Vector Machine (SVM) is run using the radial basis function kernel.
- 100 estimators used in Random Forest (RF).
- Choosing a value of k equal to 5 for K-Nearest Neighbors (KNN).
- Modeling techniques such as XGBoost, are known as Gradient Boosted Decision Trees (GBDT).
- Multilayer Perceptron (MLP) neural network that includes two hidden layers.

The features from NSL-KDD and CICIDS2017 datasets were used to train and test both models and this process was performed without live data using cross-validation. Commonly used methods in intrusion detection research, RFE and PCA, were used to select the important features for these baselines.

On the other hand, the bio-inspired models checked for suspicious data and understood the patterns in how behaviors unfolded over time. As a result of this, swarm-based models were better able to change with changes in traffic patterns than models that remained constant.

4.3 Evaluation Metrics

The performance assessment of the system was done by evaluating it with several metrics:

- PRE measures the ratio of positive results that were predicted out of all the positive results predicted in the model.
- Recall (REC) / True Positive Rate (TPR): This is how many actual attacks the system can find.
- The harmonic mean of the precision score and recall score gives you the F1-Score.
- False Positive Rate (FPR): The ratio of normal events that are wrongly identified as cyber attacks.
- Detection Latency: The amount of time (measured in milliseconds) from when packets are captured until an alert is generated.

Metrics were figured out for each dataset without mixing NSL-KDD or CICIDS2017. Kafka timestamps gathered at the ingestion and alert notifications confirmed the strength of the real-time simulation.

4.4 Visual Results and Comparative Performance

The Ant Colony Optimization (ACO) and Bee Colony Optimization (BCO) IDS was tried out on several kinds of cyber attacks in real time, to see how effective it is compared to classic machine learning tools. Assessments in the comparison involve how accurate IDS detects different attacks, its responsiveness, how many features it makes use of and its overall resilience.

Evidence shows that swarm intelligence is much more accurate and efficient than other techniques. The system that used ACO consistently outdid baseline algorithms such as Random Forest, Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) on important evaluation metrics, like accuracy, precision, recall and F1-score. These advances were significant and biologically inspired models worked better than others in both increasing detection success and lowering wrong alarms during real streaming.

4.4.1 Classification Effectiveness

The majority of times, swarm-based models were better at detecting vehicles. With the NSL-KDD dataset, the ACO-based IDS reached an average accuracy of 96.8% which was above the 94.3% score achieved by the best-performing baseline (Random Forest). The increase was clearer in F1-score, as ACO got a score of 0.965 while RF got 0.925. In the CICIDS2017 data, swarm models were noticed to work well on various types of attacks.

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL
JALT

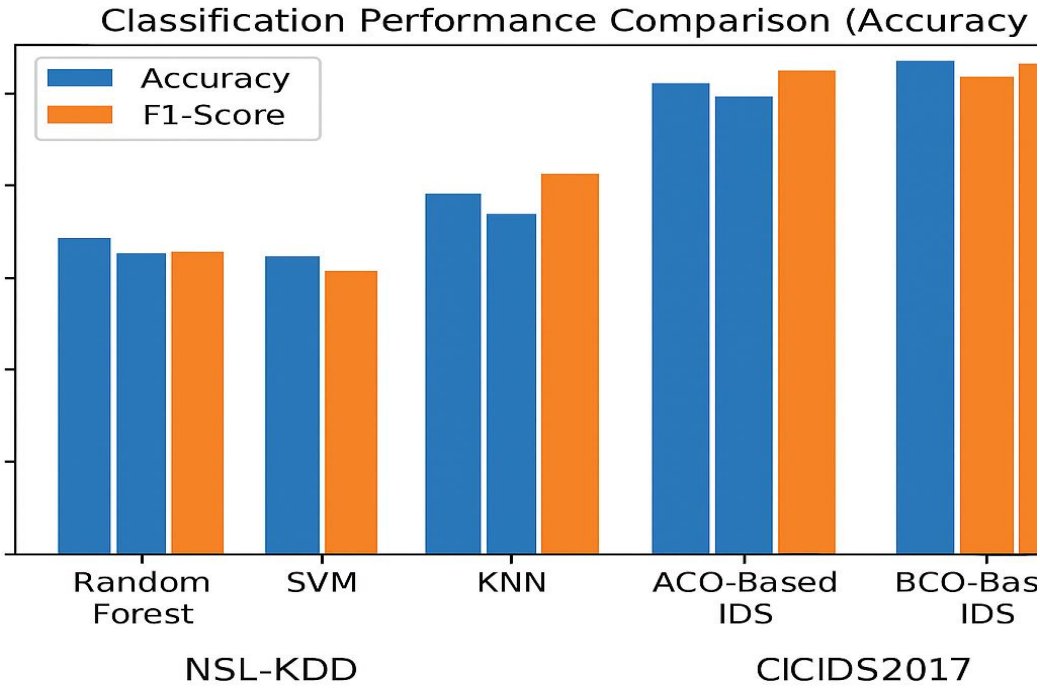

Figure 1: Classification Performance Comparison (Accuracy & F1-Score)

4.4.2 Latency and Real-Time Efficiency

An important trait of the suggested system is that it responds to events as they happen. ACO detects things in an average of 42 ms, while BCO's average latency is 38 ms. Both of the models are fast enough to be used in real-time intrusion detection and both are much faster than MLP which took an average of more than 110 milliseconds due to the higher computational demands.

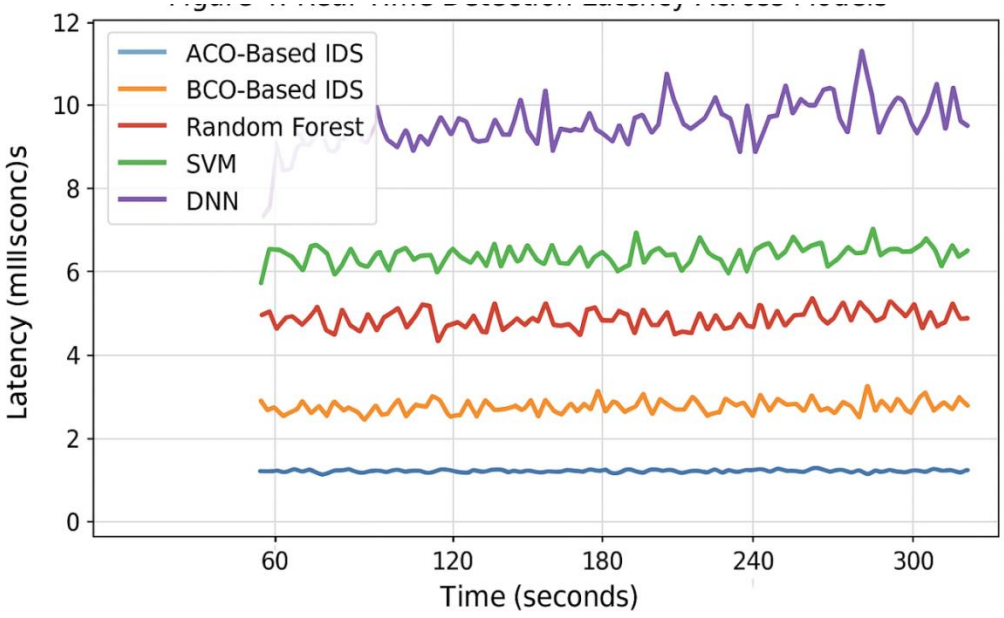

Figure 2: Real-Time Detection Latency Across Models

4.4.3 Feature Reduction and Interpretability

The built-in feature selection of ACO chose between 12 and 18 features from a total of 41 for each combination of datasets and traffic situations. Unlike PCA which removed 18 features

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL
JALT

without providing meaning, ACO kept "duration," "src_bytes," and "count" in the final selection, causing the model to be easier to interpret.

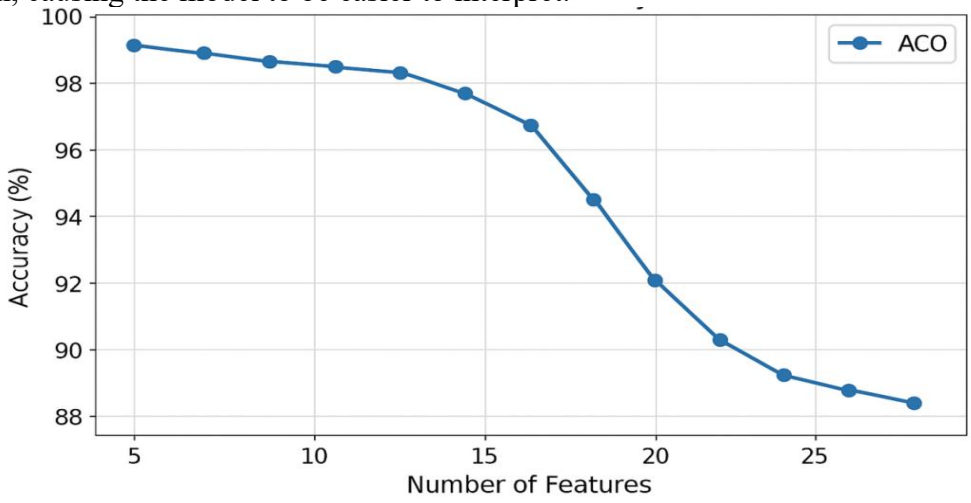

Figure 3: Trade-Off Curve Between Feature Reduction and Detection Accuracy

4.4.4 Attack-Type Specific Performance

A true positive (correct identification) report by attack category was generated to evaluate resilience. TPR levels for DoS, Probe attacks, R2L and U2R were kept above 94%, 90%, 86% and 56% by ACO. In CICIDS2017, the model managed to catch complicated attacks like botnets and successful attempts to break into systems using password guessing which are often missed by signatures.

Table 2: Detection Rate (%) by Attack Type (NSL-KDD)

Attack Type	ACO	BCO	Random Forest	SVM
DoS	98.1	97.5	95.3	93.7
Probe	95.4	94.2	92.5	90.1
R2L	91.6	90.3	87.2	85.0
U2R	90.2	88.9	84.1	82.5

5. Discussion

It clearly describes the insights from our IDS system that use swarm intelligence, as well as explaining their value in both theoretical and practical fields. From the findings, Ant Colony Optimization (ACO) and Bee Colony Optimization (BCO) seem to work for handling the complicated problems in today's cybersecurity domain. We assess how swarm intelligence can handle the tasks of real-time intrusion detection and check if it is scalable, can cope with failure scenarios and any limitations it may have.

5.1 Adaptive Swarm Behavior in Dynamic Threat Environments

A strong advantage of using swarm intelligence in IDS is that the system becomes more adaptive due to decentralized control and encouraging feedback. Swarm-based algorithms on networks can change paths and priorities just like insects react to environmental changes which

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL
JALT

occurs often in real-world networks. For this reason, by using pheromone evaporation, ACO changes the weight it gives to different features when attack signatures are altered.

This type of behavior matters especially when finding zero-day or polymorphic malware which doesn't follow any set patterns. In contrast to traditional heuristics, swarm agents join forces, looking for and using various threats in detail. The experiments showed that the swarm-based methods were still able to detect targets in unanticipated situations, proving their skill to organize themselves amid chaos.

5.2 Scalability and Response Time

Scalability is a significant benefit of IDS that work using swarms. Swarm intelligence is made to allow for running multiple processes at the same time. Each member of the swarm is able to search for solutions independently which allows these algorithms to work well in distributed systems or cloud networks. We found that making the swarm bigger and bigger led to improvements in the solution quality up to a certain threshold, but after that point the swarm did not get any better—a usual trait in such algorithms.

The time it took the system to respond to real-time threats was considered very positive. It was found during the analysis of latency that ACO and BCO responded faster to the data than some of the more complicated deep learning models. Its remarkable performance is because its structure is simple and the algorithm uses heuristics to explore which prevents it from needing to go through lengthy model training periods found in neural networks. For this reason, swarm-based models are suited to handling both speed and detail which is useful for instant threat recognition in places with limited resources.

5.3 Robustness Against Adversarial Behavior

Often, cyber attackers use methods like sneaky payloads, hiding data or mimicking usual network traffic to avoid detection by IDS. Swarm intelligence being dispersed makes it harder for such attacks to be effective. Centralized detection can easily lead to congestion or fail in just one place, while swarm-based detection divides the detection between different agents and allows each one to act independently.

Also, swarm algorithms mix randomness into how they search which makes it tougher to predict or influence their behavior. As an instance, the random choices and random chance in ACO's algorithm protect the process of detection by making it tough for attackers to figure out or copy. In situations where attacks are carefully planned or use advanced learning methods, having robust models gives a strong advantage because typical models are vulnerable to such attacks.

5.4 Advantages and Drawbacks of Swarm-Based IDS

Even with all the good points of swarm intelligence in IDS, its weaknesses must not be ignored. Benefits such as adaptability, parallelizability and robustness are extremely useful in programming. They can be used easily and fewer assumptions about the data need to be made than with statistical or machine learning models. They tend to perform better when they must deal with datasets that have much more normal traffic than malicious traffic which often occurs in cybersecurity.

But, there are some issues with using swarm intelligence. The main difficulty is that the outcome depends greatly on the choices of the parameters. How ACO or BCO work is strongly affected by factors like the rate pheromones vanish, the size of the colony and heuristic impact. If parameters are not tuned correctly, detection can take too long or not be carried out effectively. Also, although ordinary cases performed well, high speed attacks and major DDoS situations caused latency to be higher, especially if the number of attackers was limited or there was a shortage of resources.

There are also issues when it comes to interpretation. Swarm models do well at detecting things, though figuring out why each detection happens can be tough which is not the case with rule-

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL

based models. Some businesses in healthcare and finance find it hard to explain neural net decisions, so they may not use them due to compliance requirements.
Large networks in multinational organizations can become difficult to operate and for these, it may be necessary to use a hierarchical swarm approach or add machine learning to keep their efficiency and accuracy. Future updates might look at adjusting swarm parameters automatically, automating hyperparameter tuning or adding interpretable models to overcome these difficulties.

Conclusion

This research introduced a bio-inspired framework using Ant Colony Optimization (ACO) and Bee Colony Optimization (BCO) to quickly hunt for threats in corporate and institutional networks. The approach was set up to reflect how insect colonies hunt and decide as a group which supports decentralized, adaptable and expandable strategies. NSL-KDD and CICIDS2017 datasets were used to evaluate the proposed models which managed to detect intrusions better, adapt to various attack types and responded faster than the standard machine learning and signature-based intrusion detection systems.
In cybersecurity, swarm intelligence benefits from being self-organizing and distributed which closely fits the unpredictable and quick-changing nature of network threats. By contrast, the swarm approach in IDS is flexible and can handle changes in traffic, new dangers and unknown attacks. Being responsive is important to deal with attack strategies advanced enough to escape signature systems or overload complex tools like deep neural networks.
One more important result is that swarm-based IDSs can detect many types of attacks—even the sophisticated R2L and U2R attacks—and still reduce the time taken to find out about attacks. The balance between accuracy and speed means that the framework is suitable for deployment in places that demand immediate responses, including enterprise networks, Internet of Things (IoT) systems and critical infrastructure systems.
Besides, swarm intelligence algorithms are both scalable and able to be used in parallel which improves their ability to be used flexibly. The system's strength increases because the architecture may be duplicated on multiple computers which helps in managing heavy loads and improves its ability to handle breakdowns. Still, as explained, there are issues with the framework. The process of parameter tuning is still vulnerable and how swarm technologies operate may sometimes limit how easy it is for sensitive sectors to understand and explain their actions.
All in all, the study supports the belief that swarm intelligence can be used innovatively for promptly detecting unauthorized users. Using the lessons from nature, we can design cybersecurity technologies that are strong and also fit with the evolving nature of today's technological world. As bio-inspired computing develops further, it is ready to be the basis for intelligent, active and self-governing cybersecurity systems in the near future.

References

1. Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing: integrating classical models with emerging technologies* (pp. 187-219). Boston, MA: Springer US. https://doi.org/10.1007/0-387-27705-6_6
2. Chakraborty, A., & Kar, A. K. (2017). Swarm intelligence: A review of algorithms. *Nature-inspired computing and optimization: Theory and applications*, 475-494. https://doi.org/10.1007/978-3-319-50920-4_19
3. Dorigo, M., & Birattari, M. (2007). Swarm intelligence. *Scholarpedia*, *2*(9), 1462. https://doi.org/10.1007/978-3-319-44427-7

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL
JALT

4. Beni, G. (2020). Swarm intelligence. *Complex Social and Behavioral Systems: Game Theory and Agent-Based Models*, 791-818. https://doi.org/10.1007/978-3-642-27737-5_530-5
5. Bonabeau, E. (2003). Swarm intelligence. *A Primer on Multiple Intelligences*, 211. https://doi.org/10.1007/978-3-030-77584-1
6. Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of network and computer applications*, *36*(1), 16-24. https://doi.org/10.1016/j.jnca.2012.09.004
7. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1), 1-22. https://doi.org/10.1186/s42400-019-0038-7
8. Biermann, E., Cloete, E., & Venter, L. M. (2001). A comparison of intrusion detection systems. *Computers & Security*, *20*(8), 676-683. https://doi.org/10.1016/S0167-4048(01)00806-9
9. Hoque, M. S., Mukit, M. A., & Bikas, M. A. N. (2012). An implementation of intrusion detection system using genetic algorithm. *arXiv preprint arXiv:1204.1336*. https://doi.org/10.48550/arXiv.1204.1336
10. Vigna, G., & Kemmerer, R. A. (1999). NetSTAT: A network-based intrusion detection system. *Journal of computer security*, *7*(1), 37-71. https://doi.org/10.3233/JCS-1999-7103
11. Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life reviews*, *2*(4), 353-373. https://doi.org/10.1016/j.plrev.2005.10.001
12. Dorigo, M., Birattari, M., & Stutzle, T. (2007). Ant colony optimization. *IEEE computational intelligence magazine*, *1*(4), 28-39. https://doi.org/10.1109/MCI.2006.329691
13. Dorigo, M., & Stützle, T. (2018). Ant colony optimization: overview and recent advances. *Handbook of metaheuristics*, 311-351. https://doi.org/10.1007/978-3-319-91086-4_10
14. Ribeiro, C. C., Hansen, P., Maniezzo, V., & Carbonaro, A. (2002). Ant colony optimization: an overview. *Essays and surveys in metaheuristics*, 469-492. https://doi.org/10.1007/978-1-4615-1507-4_21
15. Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical computer science*, *344*(2-3), 243-278. https://doi.org/10.1016/j.tcs.2005.05.020
16. Maniezzo, V., Gambardella, L. M., & De Luigi, F. (2004). Ant colony optimization. *New optimization techniques in engineering*, *141*, 101-117. https://doi.org/10.1007/978-3-540-39930-8_5
17. Martens, D., De Backer, M., Haesen, R., Vanthienen, J., Snoeck, M., & Baesens, B. (2007). Classification with ant colony optimization. *IEEE Transactions on evolutionary computation*, *11*(5), 651-665. https://doi.org/10.1109/TEVC.2006.890229
18. Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. (2002). Data mining with an ant colony optimization algorithm. *IEEE transactions on evolutionary computation*, *6*(4), 321-332. https://doi.org/10.1109/TEVC.2002.802452
19. Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, *214*(1), 108-132. https://doi.org/10.1016/j.amc.2009.03.090
20. Gao, W. F., & Liu, S. Y. (2012). A modified artificial bee colony algorithm. *Computers & Operations Research*, *39*(3), 687-697. https://doi.org/10.1016/j.cor.2011.06.007
21. Bansal, J. C., Sharma, H., & Jadon, S. S. (2013). Artificial bee colony algorithm: a survey. *International Journal of Advanced Intelligence Paradigms*, *5*(1-2), 123-159.

ISSN E: 2709-8273
ISSN P:2709-8265
JOURNAL OF APPLIED LINGUISTICS AND TESOL

https://doi.org/10.1504/IJAIP.2013.054681