INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Toward Enhancing Malware Detection Using Practical Swarm Optimization in Honeypot

Heba Othman[1] **Mwaffaq Abu AlHija**[1]* **Mohammad A. Alsharaiah**[1]

[1]*Al-Ahliyya Amman University, Amman, Jordan*
* Corresponding author's Email: *m.abualhija@ammanu.edu.jo*

Abstract: Malware attacks have become a pressing concern in the domain of computer and communication systems, posing significant threats to data security and privacy. A practical approach is represented aiming to enhance the malware detection process with the help of Honeypot. A hybrid model of particle swarm optimization (PSO) integrated with Fuzzy_KNN algorithms is used in this research. Numerical simulations and mathematical analysis are conducted after developing numeric codes of this scheme. The performance and practicality are examined via these evaluation metrics including accuracy, precision, recall, and F1-score. Based on the numerical investigations, the findings confirmed satisfying performance measures of the hybrid model. The FuzzyKNN algorithm does attain the most remarkable effectiveness, achieving accuracy between 99.95% and 99.97%. This model employs a premium method of neighbourhood voting with an element of fuzziness and excels in large or complex datasets where patterns may emerge based on instance similarity.

Keywords: Machine learning, Particle swarm optimization, FuzzyKNN, Performance evaluation, Enhancing malware detection.

1. Introduction

Worldwide, computers, gadgets, communication technologies, and network infrastructure have been uniformly created and advanced. The explanations are tied to the vast digital development and increasing telecommunication wealth. These elaborations have been done in most international cities according to Volkodaeva [1], B. Bygstad [2], M. Attaran [3], and M. Knell, [4], this rise provided different significant advantages, including big data (BD) with diverse information resources available on the web and countless industrial and engineering improvements. At the same time, it is necessary to identify the significant negatives or anticipated drawbacks of this exceptional improvement in information technology that transpired at a wide scale to make the appraisal of technical innovation more realistic and just. An example of the considerable negatives that have been extensively observed in this technical advancement is the introduction of big aggressive internet attacks and cyber threats.

This part is owing to this tremendous international digitalization. Consequently, this development leads to a collection of issues and significant impediments in safeguarding databases, whether for people, corporations, or governments, hurting their data security guided by M. Seete [5, 6], Gangwar & Narang [7], Cybersecurity refers to the process of securing computer infrastructure from intrusion, theft, and harm caused by digital techniques. This entails adopting efforts to prevent and mitigate online threats including hacking, phishing, and malware. Information security is a crucial aspect of today's technology infrastructure, needing both technical and non-technical measures to secure the privacy, validity, and accessibility of data. Furthermore, information security, or information safety technology refers to the process of securing information through limiting information risk.

In the ever-evolving landscape of cybersecurity, the threat of malware remains a pervasive and constantly mutating challenge. Malware, short for malicious software, encompasses a wide array of harmful programs designed to infiltrate, damage, or gain unauthorized access to computer systems.

Traditional methods of malware detection often face limitations in effectively identifying new and sophisticated malware strains. Cyber attackers continuously refine their tactics, employing techniques that evade conventional security measures. In this context, innovative approaches rooted in advanced computational intelligence have emerged as a promising frontier.

One such approach is practical swarm optimization (PSO), a metaheuristic algorithm inspired by the collective behavior of social insects, such as bees and ants. PSO leverages the power of swarm intelligence, where individual agents collaborate in a decentralized manner to achieve a common goal. In recent years, researchers have explored the application of PSO in various domains, including data mining, optimization, and machine learning. By harnessing the collective intelligence of a swarm, PSO demonstrates the potential to enhance the accuracy and efficiency of malware detection systems.

This research delves into the integration of practical swarm optimization within the realm of honeypots, specialized decoy systems designed to lure cyber attackers into revealing their techniques and methodologies. By combining the adaptive nature of PSO with the honeypot environment, this study aims to enhance the ability to identify and classify novel malware strains. The collaboration between computational intelligence and cybersecurity holds the promise of creating more resilient and adaptive defense mechanisms, thereby bolstering the overall cybersecurity posture of organizations and individuals in the face of evolving cyber threats

The phrase “information risk management” refers to the activity of minimizing the probability that sensitive data may be accessed, exploited, revealed, intercepted, deleted, destroyed, analyzed, recorded, or altered by unauthorized personnel. It also comprises efforts to mitigate the impact of the circumstance by [8].

Data breaches, in which private information like as names, addresses, and credit card numbers are taken or published, are often seen as the most bothersome sort of cyber-attack nowadays. One of the most common cases of data breaches that may show the foundations of cybersecurity is the Malware cyber attack (MCA).

It is worth highlighting that conventional antivirus systems and signature-based detection methods are less successful against current malware due to the latter’s increased complexity and sophistication by B. Lutkevich [9]. Depending on this, computer professionals, internet scientists, cyber security experts, and high-knowledge technological specialists have conducted extensive investigations and research to help create effective solutions and feasible approaches that might address those obstacles and detect different MCAs with significant levels of accuracy and performance. It is worth emphasizing that conventional antivirus systems and signature-based detection approaches are less successful against modern malware due to the latter’s increased complexity and sophistication according to [10]. Depending on this to help create effective solutions and feasible approaches that might address those obstacles and detect different MCAs with significant levels of accuracy and performance. Two examples of those favorable tactics are fuzzy logic (FL) and particle swarm optimization (PSO). Computer professionals, have conducted extensive investigations and research to help create effective solutions and feasible approaches that might address those obstacles and detect different MCAs with significant levels of accuracy and performance.

It is worth emphasizing that conventional antivirus systems and signature-based detection approaches are less successful against modern malware due to the latter’s increased complexity and sophistication presented by M. Akhtar [10]. Internet scientists, cyber security experts, and high-knowledge technological specialists have conducted extensive investigations and research to help create effective solutions and feasible approaches that might address those obstacles and detect different MCAs with significant levels of accuracy and performance. Two examples of those advantageous strategies are fuzzy logic (FL) and particle swarm optimization (PSO). Some of the benefits of the fuzzy logic model include the supply of inherent flexibility and a smooth control function (output control), where noise-free inputs, the removal of fixation, and flexible programming for safe failure when the feedback sensor fails or is destroyed are all supplied. Secondly, FL provides a rule-based user approach in which transcending the intended control system may be done. Thirdly, FL supplies a large selection of control outputs and feedback mechanisms that help generate greater cost-effectiveness and reliable performance for the solutions related to different issues according to C. Li, [11] and M. Bhagwat [12]. Furthermore, researchers and cyber security professionals have identified another way that can be capable of capturing attackers by building virtual traps, which is known as a honeypot. Thus, a substantial percentage of malware threats may be

Received: September 20, 2023. Revised: November 1, 2023.

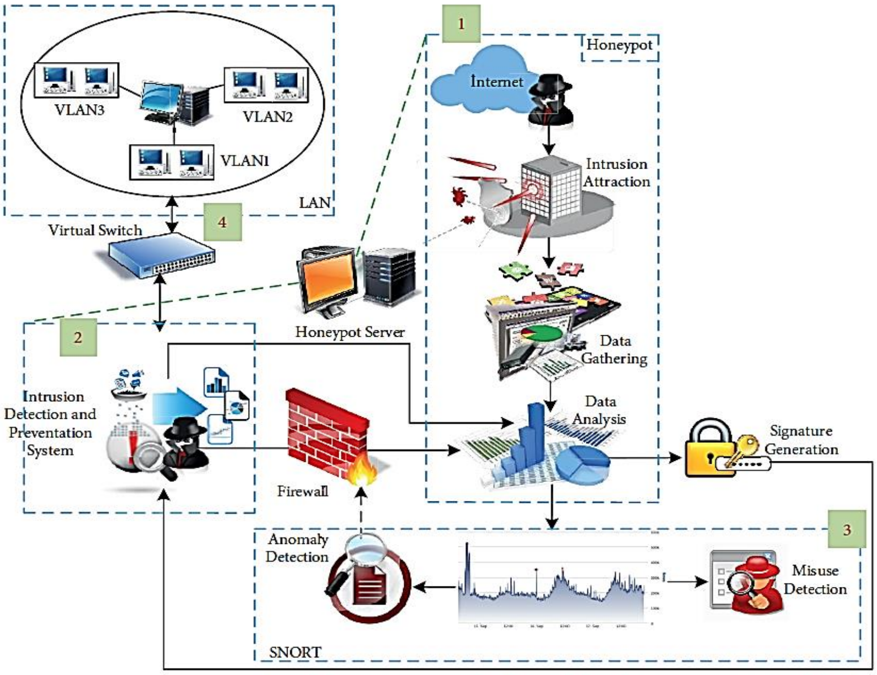

Figure. 1 The principal architecture of the honeypot approach employed for detecting Different cyber threats by P. Yadav [13]

minimized. Fig. 1 depicts the primary structure and working principles of the honeypot system applied for the detection of cyber threats.

Honeypots detect malware. Researchers can identify and prevent malware by monitoring its behavior in this honeypot. Honeypot malware research and classification needs human analysis and many tools and procedures, making it time-consuming and resource-intensive by M. Baykara [14]. Optimize this. Swarm optimization has handled image recognition, network routing, and ML challenges. These approaches accelerate and enhance honeypot malware detection and categorization [15] recommends fuzzy logic for smart ML (FL). Variable values are real integers. FL quickly identifies malware [16]. Thirdly, KNN is useful. AI helps this numerical paradigm work. Several research shows this strategy enhances malware detection. Academics and computer engineers can boost cyber security with those three methods. Multiple ML algorithms improve accuracy and efficiency. Our hybrid PSO-FuzzyKNN model detects malware better. Higher honeypot measures. Learn three key numerical research analysis and interpretation concepts.

Three ideas: particle swarm optimization (PSO): A sophisticated algorithm that employs iterations to solve problems or find malware infections and other cyber threats according to İ. Atacak [17].

- Honeypots secure Internet traffic. It may replicate cyber hazards to lure attackers and others who seek to destroy.
- Databases and steal network data through viral attacks guided by Phommixay, S., [18].
- Artificial intelligence (AI): Numerical programs and computer systems use training and learning to improve their performance, dependability, and efficacy in solving network problems or detecting malware threats and cyberattacks utilizing intelligent algorithms by [19]. Fuzzy logic this unusual ML approach lets variables have positive or negative real integer values other than 0 and 1. Uncertain values use partial truth. KNN: Practical non-parametric regression and classification.
- Closest training data is used. KNN regression/classification choices output.
- FL-KNN mix. When FL or KNN fail, this combination improves efficacy and speed. Both approaches support the new ML model. Computer and communication malware has plagued wealthy and impoverished nations for decades. Malware assaults have damaged personal PCs and laptops, business networks, databases, and network flaws, weakening consumers' faith in network and internet databases.

- Database accesses utilizing distinct approaches. New lures may catch network and database malware attackers. Honey pots innovate.
- Honeypots mimic cyber threats to catch criminals [20], [21], [22].
- ML and AI malware detection methods were examined.

Nonetheless, researchers and scholars in the available literature have investigated the detection of cyber malware threats depending only on PSO or fuzzy logic alone Tian, W. [23]. Therefore, the current study is conducted to implement PSO and FuzzyKNN algorithms along with the honeypot to offer a high-performance, powerful, and reliable detection process of malware attacks that may harm the comfort and ability of internet users. The work is implemented via optimization using numerical code developed and run in Python software. [24]

1.1 Inadequacies of traditional malware detection:

Traditional signature-based malware detection methods rely on predefined patterns and signatures to identify malicious software. While effective against known malware, these techniques falter when encountering previously unseen or mutated strains. Moreover, behavioral analysis methods, which observe the actions of programs in a controlled environment, face challenges in accurately distinguishing between benign and malicious behavior. As malware becomes more sophisticated, it increasingly evades these conventional detection mechanisms, by Tiwari, S. [25]

1.2 Challenges in honeypot environments:

Honeypots, decoy systems designed to attract cyber attackers, serve as valuable tools for understanding attack patterns and gathering threat intelligence. However, existing honeypot systems often struggle to adapt swiftly to emerging malware behaviors. The challenge lies in creating honeypots capable of not only luring attackers but also effectively detecting and classifying previously unknown and Polymorphic malware strains. The inefficiency of traditional detection methods within honeypot environments underscores the need for innovative solutions. Presented by S. Aljawarneh [26], Lin& Wang [27]

1.3 The role of practical swarm optimization (PSO)

Practical swarm optimization (PSO) emerges as a potential solution to these challenges. PSO, inspired by the collective intelligence of social insects, offers an adaptive and self-learning mechanism. By harnessing the power of swarm intelligence, PSO has the potential to optimize the detection process within honeypots. The challenge here is to effectively integrate PSO into honeypot systems, enabling real-time adaptation to the dynamic nature of malware threats. This integration requires addressing the intricacies of swarm behavior, ensuring efficient information sharing, and determining optimal parameters for adaptive decision-making according to S. Aljawarneh [26]

This research aims to address the deficiencies in existing malware detection methodologies within honeypot environments. Specifically, the objective is to design and implement a system that seamlessly integrates practical swarm optimization into honeypot technology. The goal is to enhance the accuracy, speed and adaptability of malware detection and classification by P. Wang [28], especially concerning previously unseen and evolving malware strains. By achieving this objective, the research seeks to contribute significantly to the advancement of cybersecurity practices, enabling organizations to proactively combat the ever-changing landscape of cyber threats.

The available literature, especially in the last two decades, witnessed significant growth in the number of articles and peer-reviewed papers published in different journals and discussed numerous advantages and positive impacts of ML and AI in detecting different cyber threats on the internet, servers, and networks. At the same time, the benefits of the PSO approach were analyzed but in special cases and limited applications.

In addition, it is observed that the available literature focused only on one approach of malware detection, like investigating the Beneficial impacts of PSO and FuzzyKNN models besides honeypots. Even the global literature did not analyze the relevant benefits of utilizing the three concepts (which are the PSO, FuzzyKNN, and the honeypot system) in one research. [29]

2. Literature review

2.1 Substantial merits of FuzzyKNN in recognizing cyber and malware threats

Atacak, [16] wrote about the importance of ML models like the FuzzyKNN model in cybersecurity. When used with clever ML models, honeypots improve cyber issue detection. They said the internet is essential to modern life. Professionals, academics, and corporations use digitalization. However, deep/dark web cyberattacks have plagued the worldwide internet network. They used web content and in-depth content analysis. Crawler frameworks were researched to speed up and secure searches. The search yielded numerous forms of data that were added to their database. Database categorization determined site maturity. Classification using Fuzzy-KNN. Fuzzy-KNN classified database crawling framework results. The crawling framework produced URLs, page information, and more. Crawling data was compared to a sample. Fuzzy-KNN may classify web pages based on the sample data word value.

Limited test data. Thus, more data was needed. An improved crawler framework can expedite results at higher web levels when the Tor browser can be used but the crawler framework cannot. Wang, P. [28] performed a comprehensive analysis of fuzzy techniques, including the FuzzyKNN model, in detecting DDoS attacks and network traffic irregularities. Their lengthy analysis showed anomalous intrusion detection systems may seek unique behavior rather than security issues. Fuzzy data mining and statistical methods reduce intrusion detection uncertainty with anomaly detection. Fuzzy logic studied network traffic anomalies and DDoS attacks.

Fuzzy techniques examined DDoS attacks and network anomalies. Fuzzy network anomaly detection systems employ classifiers, feature selection/extraction, statistical and clustering approaches, and others. Discussed honeypots and fuzzy model augmentation. They investigated network anomaly detection. Also, practical proposals and significant research routes were made. J. Hwang [24] Evaluated KNN and FuzzyKNN algorithms for cyber threat identification in mobile Ad-Hoc network

2.2 Fuzzy logic's essential key strengths

FL is a significant ML model that helps businesses and researchers make the best choice and solve many problems: Intrinsic resilience, smooth output control, and fixation-free inputs:

1. The process of user-defined rules and overriding the target control system. The system performance can be improved since it can be changed.
2. Offering multiple control outputs and various feedback inputs.
3. The employment of the rule-based operation through which inputs of reasonable numbers, such as (1-6 or more), and diverse outputs, like (1-4) can be produced.
4. The capability of managing non-linear systems that would be difficult to manage mathematically, helping measure feasibility for automation.
5. The FL model may also create sensors with similar guiding principles. Pre-deployment change rate variables are unnecessary. Sensor data suffices. Sensors are cheaper and system profitability increases.

2.3 Critical benefits of PSO with honeypot system to detect malware threats

Aljawarneh and Al-Betar [25] examined the crucial role and functional contributions of a honeypot system-PSO malware detection model. PSO-optimized feature selection for an effective detection model. The honeypot also caught malware activity from potential attackers. The model is tested with harmful and benign samples. The model detected malware with a high detection rate and low false-positive rate, according to their numerical study. S. Aljawarneh [26] led a study on intelligent approaches for identifying online and network malware. The authors tuned fewer variables using the PSO technique. However, they evaluated another new malware tracking method developed by scholars in recent decades. Honeypot technique. This unique method intelligently caught malware attackers, improving the PSO's reliability, performance, and efficiency when the honeypot system was utilized to identify malware. Their technique, which used PSO and the honeypot system, had reduced false-positive rates and higher detection rates for benign samples and varied datasets. C. Lin, [27] applied the principles of various network-attached systems to imitate targets and ensnare cyber-attacks against enterprises' and institutions' networks. Honeypots are target tools. Moreover, the scientists used the feature selection method, which was accomplished by the PSO algorithm. An incremental feature selection framework made feature selection more effective.

2.4 Implementing fuzzy logic and honeypot concept in detecting malware threats effectively

Kiran and Khandelwal [28] examined how fuzzy logic and honeypots identify cyber viruses. Honeypot and fuzzy logic numerical analysis detected malware. Their honeypot captured malware. Numerical simulations and quantitative analysis revealed their detection strategy was more accurate and efficient. Detection enhanced. Gupta and Dutta [29] explored honeypot systems and fuzzy logic ML models. Unique methods found multiple malware families. Their article described honeypot-and-fuzzy-logic-based active malware detection. Honeypots captured malware. Fuzzy logic ensured detection. Their malware detection method surpasses cutting-edge algorithms and ML models. Jain and Sood [30] demonstrated honeypot and fuzzy logic malware detection. Honeypots detected malware. Mathematical modeling found numerous malware concerns. ACC claimed superiority. Fuzzy logic and honeypots identify cyber infections, according to Wang, P. [28]. Malware was detected through honeypot and fuzzy logic numerical analysis. Malware was found in their honeypot. Their detection method was more accurate and efficient, according to numerical simulations and quantitative evaluations. Improved detection.

Honeypot systems and fuzzy logic ML models were examined by T. Kiran [29] several malware families were found using novel methods. Honeypot-and-fuzzy-logic active malware detection was discussed in their article. Malware was found in honeypots. Detection was reliable with fuzzy logic. They outperform cutting-edge algorithms and ML models in malware threat identification. Honeypot and fuzzy logic malware detection by [30]. They used a honeypot to identify malware. Malware problems were identified using mathematical modeling. It outperformed others, according to ACC. They also found out how altering elements may affect their strategy's efficiency and set certain requirements and constraints. Arias. [31] Used a honeypot system to test fuzzy logic models for computer attack detection. Honey nets are used nowadays to detect network vulnerabilities and attackers. Knowledge of existing structures, technologies, and improvements is needed for effective application. Active fingerprinting attacks inject honeypot-specific network traffic. Limiting the honeypot's external connections protects this attack but renders it useless. The honeypot can be protected against fingerprinting if spotted quickly. Thus, they created a self-aware fingerprinting honeypot using D-FRI. Their minimum rule set detected continual fingerprinting threats without matching. Their numerical analysis demonstrated that D-FRI responded to network conditions and provided a dynamic set of rules, boosting detection precision, reliability, and efficiency.

2.5 Utilizing PSO and fuzzy logic for reliable malware threat identification

Dharshini and Tamilarasi [32] conducted a hybrid study of two ML algorithms. PSO and fuzzy logic ML were utilized. The detection model was fine-tuned using PSO after training on multiple datasets. The researchers assessed their technique using various performance metrics.

Compared to standard methods, their PSO-fuzzy logic model was accurate and efficient. It detected more and erred less. Bhagat and Bajaj [11] efficiently identified network malware using two numerical ML methods. The study employs fuzzy logic and PSO. PSO chose. Fuzzy logic analyzed PSO optimization. Model performance was measured. Simulations and numerical research improved malware detection. Their model outperformed ML and advanced algorithms. A hybrid method evaluated ML algorithms. PSO and fuzzy ML were used. After training the detection model on many datasets, PSO adjusted its parameters. Many performance factors assessed the researchers' technique. The research found the PSO-fuzzy logic model more accurate and efficient. Errors decreased. Bhagat and Bajaj [11] successfully identified network malware using two numerical ML approaches.

2.6 Review of PSO’s critical principles

PSO it's in swarm intelligence metaheuristics. Simulating swarm particle social behavior yields the optimal optimization solution by M. Dharshini [33]. A particle swarm is initialized in a multidimensional search space. Particles have location and velocity vectors that describe their current solution, direction, and speed. The optimization problem's objective function determines particle fitness. The swarm's global best and each particle's pbest determine its position and velocity (gbest). Position, velocity, and optimal places are updated. Fig. 2. Shows how new particle locations are evaluated for fitness until a termination criterion is met, such as a limited number of iterations or a satisfactory solution.

The equation for the velocity and position of each particle in PSO can be expressed as:

$$v_i(t+1) = wv_i(t) + c_1 r_1 (p_{besti} - x_i(t)) + c_2 r_2 (g_{best} - x_i(t)) \tag{1}$$

And:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

Where vi (t) and xi (t) are the velocity and position of the i^{th} particle at time t, w is the inertia weight, c_1, and c_2 are the acceleration coefficients, r_1 , and r_2 are random numbers, pbest, i is the personal best position of the ith particle, and gbest is the global best position of the swarm.

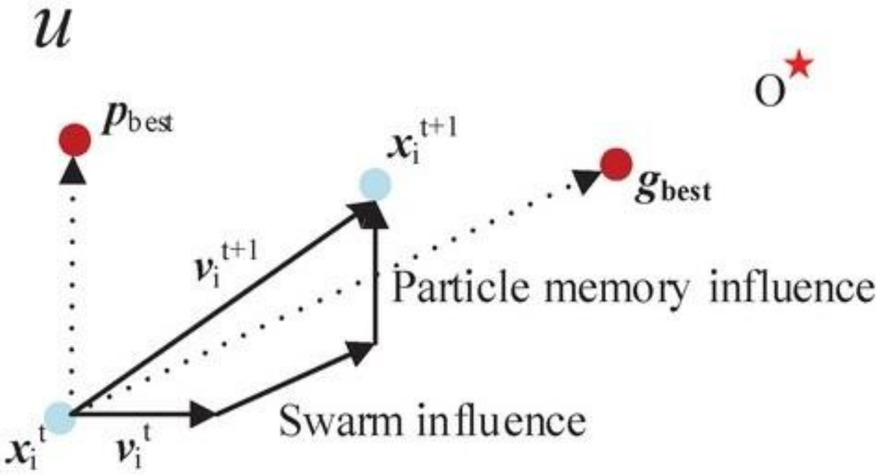

Figure. 2 Particle swarm optimisation [38]

2.7 Detection of malware through the implementation of PSO using honeypot

Abraham and Chengalur-Smith [39] stated social engineering still spreads malware. They studied social engineering malware occurrence. Modeled social engineering malware. They examined typical attack paths to explain their prevalence and endurance. Psychological and technological methods fool computer users into unleashing infection and evading security.

El-Ghamry [36] found that IoT applications have made malware a sophisticated danger. Without proper protection, hackers might steal a lot of sensitive and confidential data. This necessitates stronger network security techniques to monitor network traffic and promptly identify malicious activity. This paper proposes an efficient ML image-based IoT malware detection approach using network traffic photographs. This method used ant colony optimizer (ACO)-based feature selection to get the fewest features while improving support vector machines (SVMs) classifier performance (i.e., the results of malware detection). PSO also updated the SVM settings of kernel functions. On a publicly accessible dataset, the F1 score was 95.26 percent, accuracy 95.56 percent, recall 96.43 percent, and precision 94.12 percent.

2.8 Fuzzy logic rationale in detecting cyber and malware threats

Novae [37] led an investigation to establish the major benefits of fuzzy logic algorithms with long short-term memory (LSTM) in identifying cyber threats in software-based network systems. Computer networks are now complicated and ever-changing, the researchers said. Due to this, setting up and maintaining the framework is difficult. Thus, new networking paradigms like software-defined networks (SDN) are needed to abstract network design plans so the control plane may function independently from the data plane. Traditional network vulnerabilities exist in SDNs. LSTM and fuzzy logic were used to identify and prevent DDoS and port scan attacks in SDN scenarios. Characterization, anomaly detection, and mitigation comprised their hybrid model. System tests were two. The researchers used IP flows to imitate SDN Floodlight controllers on Mininet.

The second dataset was DDoS 2019. Their research showed that their new hybrid architecture was best at network management, attack detection, and mitigation. Avkurova [41] examined how fuzzy logic detects variant cyberattacks in online networks. Despite the ubiquitous usage of communication networks, especially in crucial physical infrastructure regions, hackers and other emerging dangers may readily penetrate current information and communication systems, the authors said.

The fuzzy logic system may actively improve cyber security by identifying cyber hazards. Their numerical results suggested constructing an updated intrusion detection system using honeypot technology or other intelligent ways to detect and identify online threats. An ML model that uses honeypot network traffic data to identify and classify malware can be improved using PSO approaches. Then, fuzzy logic algorithms can optimize utilizing PSO feature selection outputs to identify malware threats. Following the overview and summary, this study examines the performance and reliability of enhanced malware detection utilizing PSO and fuzzy logic algorithms with a honeypot system.

The experimentation adhered to ethical guidelines and legal regulations, ensuring the responsible handling of malware samples and the privacy of any data collected during the study. All experiments were conducted in a controlled environment to prevent any unintended consequences or leakage of sensitive information.

3. Materials and methods

The experimentation for enhancing malware detection using practical swarm optimization (PSO) in honeypot environments was conducted in a controlled virtualized network. The setup comprised a series of high-interaction honeypots strategically deployed across different network segments to simulate diverse environments, including web servers, databases, and email servers. Realistic operating systems and applications were emulated within these honeypots to lure potential attackers.

Malware samples, both known and unknown, were collected from diverse sources, including public malware repositories, cybersecurity forums, and dark web monitoring sources. These samples formed the basis for training and testing the PSO-enhanced detection system. Additionally, real-world attack data, obtained from security incident logs and honeypot interactions, were integrated into the dataset to simulate authentic attack scenarios.

The PSO algorithm was implemented and integrated into the honeypot environment. Parameters such as swarm size, inertia weight, and acceleration coefficients were fine-tuned through iterative experimentation to optimize the detection process. The PSO algorithm was adapted to analyze patterns of honeypot interactions, focusing on distinguishing between benign and malicious activities.

Various features related to network behavior, system calls, and file characteristics were extracted from the honeypot interactions. Feature selection techniques, including information gain and correlation analysis, were applied to identify the most relevant features for malware detection. The selected features were used to create feature vectors for training the PSO-based detection model.

The collected dataset was divided into training and testing sets, ensuring a balanced distribution of malware and benign samples. The PSO algorithm was trained on the feature vectors of known malware samples to learn their patterns of behavior. Subsequently, the system's detection capabilities were evaluated on unseen data, including both known and novel malware strains, as well as legitimate interactions.

The performance of the PSO-enhanced malware detection system was evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Comparative analyses were conducted against traditional signature-based detection methods and machine learning approaches to assess the effectiveness of the PSO integration in improving malware detection accuracy and reducing false positives.

Post-experimentation, extensive data analysis was performed to interpret the results. Statistical methods and visualization techniques were employed to gain insights into the behavior of the PSO-enhanced malware detection system, allowing for a comprehensive understanding of its strengths and areas for improvement.

Dataset collection stage: The first phase is the collection of the N-BaIoT dataset this dataset comprises real traffic data infected by Mirai and BASHLITE malware. A honeypot design was employed to capture this data, utilizing several virtual machines (VMs) connected to a network cluster.

Data preprocessing stage: The raw data goes through several preprocessing steps to ensure its suitability for machine learning algorithms: incomplete entries are removed or imputed, to avoid class bias, the dataset is balanced, and then Normalization standardizes features to optimize machine learning techniques.

Machine learning algorithms require numerical inputs, therefore label encoding converts categorical variables to numbers. Model training and assessment use preprocessed dataset training and testing sets. Feature optimization: The next step is to optimize the features of the dataset to enhance the malware detection model's performance. Particle swarm optimization (PSO) is used for this purpose. PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution concerning a given measure of quality, such as log-loss on the validation dataset. This process aids in finding the best combination of features that will improve the model's predictive accuracy.

After feature optimization, a FuzzyKNN model is trained on the improved features. Fuzzy logic helps the FuzzyKNN algorithm classify datasets with fuzzy class boundaries. Model evaluation: The final step is to evaluate the FuzzyKNN model's performance using various metrics such as ACC, PRE, REC, and F1-score.

These metrics offer a comprehensive view of the model's performance, allowing for the identification of areas of improvement and future research directions.

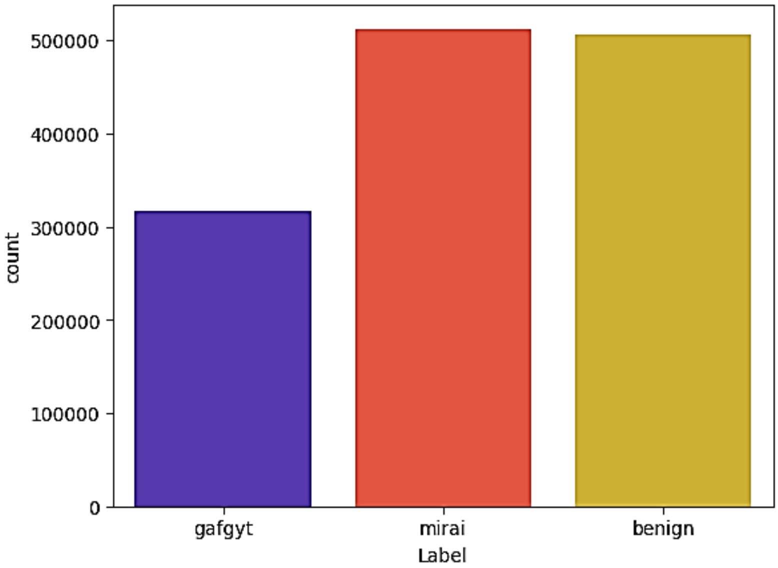

Figure. 3 Distributing data honeypots were used to capture N-BaIoT data

3.1 Data collection

The N-BaIoT dataset, created by [42], is a comprehensive collection of network traffic data from nine devices. These devices were intentionally infected with the (Mirai and BASHLITE) botnets to gather real malicious traffic data. The dataset, which includes 115 features, was collected using port mirroring and includes over 1.3 million samples. These samples are divided into three categories: Mirai attack, benign, and Gafgyt (BASHLITE) attack. This dataset is a valuable resource for cybersecurity research and comprehensive collection allows for the development of machine learning models able to distinguish different types of botnet attacks and benign traffic data, as shown in Fig. 3.

As illustrated in Fig. 4, the Bot-IoT dataset's testbed is a cluster of malicious and benign virtual machines (VMs) linked to LAN and WAN interfaces, making it a valuable resource for analyzing and modeling security risks.

3.2 Data preparation

3.2.1. Handling missing values

Shows that none of our data are null or missing. No data is superfluous. No missing values mean no imputation or data filling. To eliminate duplicate data to improve the dataset.

The dataset is missing values-free, however, feature extraction, label encoding, and data normalization are needed to optimize it for analysis.

3.2.2. Label encoding

Label encoding assigns unique numerical labels to categories to turn categorical data into numerical data. Here, "benign," "gafgyt," and "mirai" are assigned numerical values (0, 1, and 2). This transformation helps ML systems view

Categorical input as numerical data, making model fitting easier. 1 and 0 have frequencies of 828,783 and 506,384, respectively. Malware is 1 while benign is 0. The dataset is skewed, with malware traffic outnumbering benign traffic. Imbalanced datasets may influence ML model performance, therefore additional procedures may be needed.

3.2.3. Data balancing

To utilize down samplings like random under sampling, oversampling, or both to fix the unbalanced dataset. The algorithm randomly under-sampled the majority class (label 0) by picking 506,384 samples from the minority class (label 1). This created a balanced dataset with equal labels. Balancing the dataset can increase ML model performance by minimizing bias towards the dominant class and allowing the model to learn from both classes. Choose balanced data forms to train and test the FuzzyKNN classifier (illustrated in Fig. 5).

3.2.4. Data normalization

Data normalization is an essential preprocessing step in machine learning and data analysis, aimed at

Received: September 20, 2023. Revised: November 1, 2023.

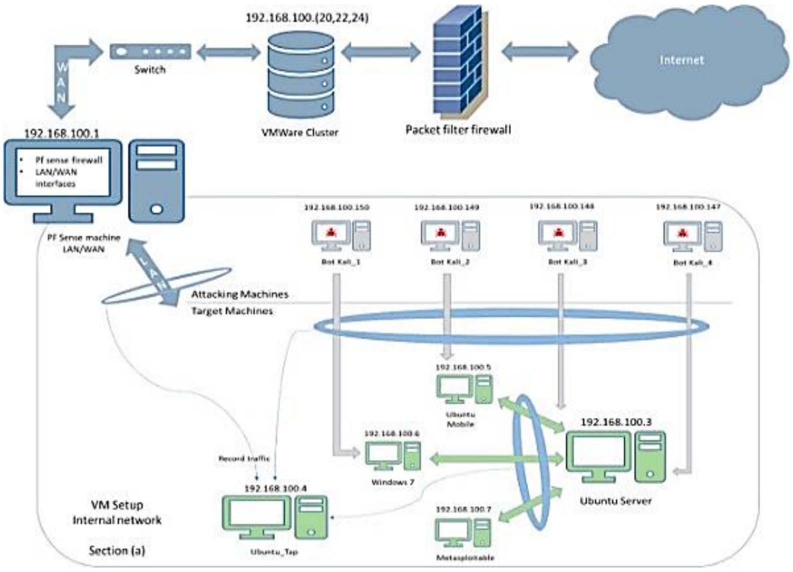

Figure. 4 Bot-IoT honeypot [42]

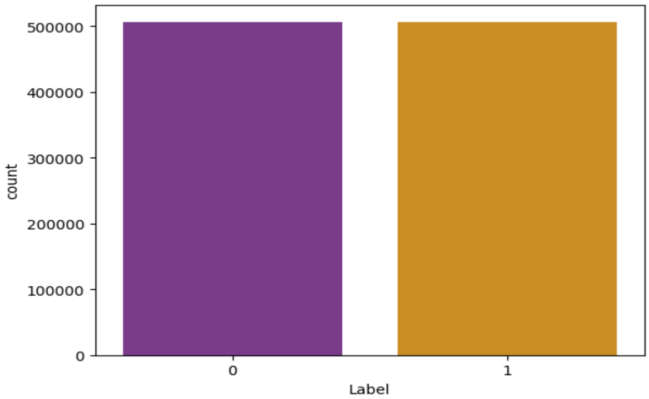

Figure. 5 Balanced data distribution

transforming the data into a standardized format. One commonly used technique for normalization is called "Min Max Scaler." This method rescales the data to fit within a range of 0 to 1. This is achieved by subtracting the minimum value of the feature and dividing it by the range of the feature.

3.3.5 Data splitting

Training and testing datasets were split. The test size option is set to 0.2, meaning 20% of the dataset will be tested and 80% trained. To eliminate ordering bias, the data was randomly mixed before splitting. Splitting the dataset lets us train and test our ML model.

3.3 Feature optimization

PSO optimizes socially using bird flocks or fish schools. Strong swarm intelligence algorithm metaheuristic. As demonstrated in Fig. 6. PSO mimics swarm behavior to find the optimal optimization solution. Launch a particle swarm into a multidimensional search space. Each particle's position and velocity vector define the solution and its direction and speed. Applying the optimization problem's objective function to each particle's position determines its fitness. The objective function evaluates a solution's limitations and objectives. Each particle's location and velocity are updated depending on its personal best (pbest) and the swarm's global best (gbest). Current position, velocity, and optimal locations inform this update. After evaluating each particle's new position for fitness, the procedure is continued until termination criteria, such as a maximum number of iterations or a satisfactory solution, are met.

The parent level of each particle in PSO is expressed as

$$vi(t + 1) = wvi(t) + c1r1\big(pbesti - xi(t)\big) + c2r2(gbest - xi(t)) \quad (3)$$

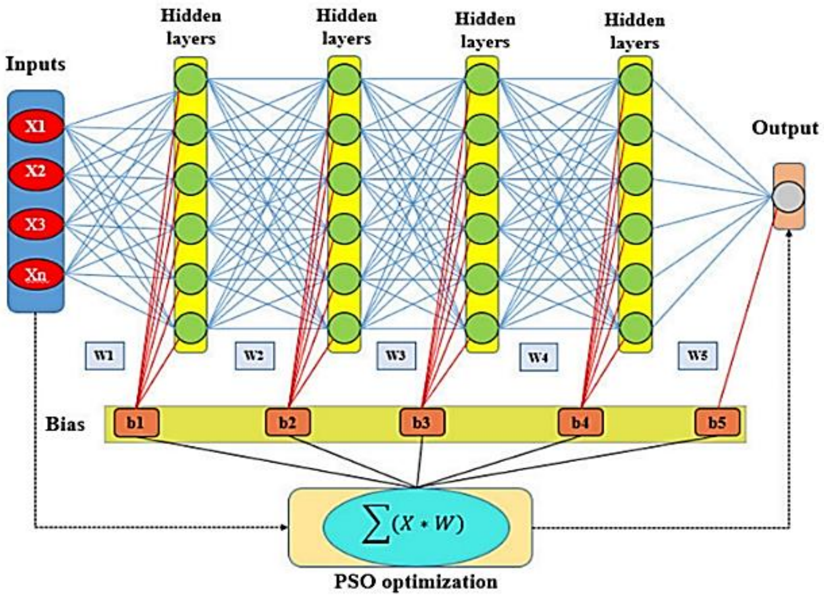

Figure. 6 PSO’s working principales. [46]

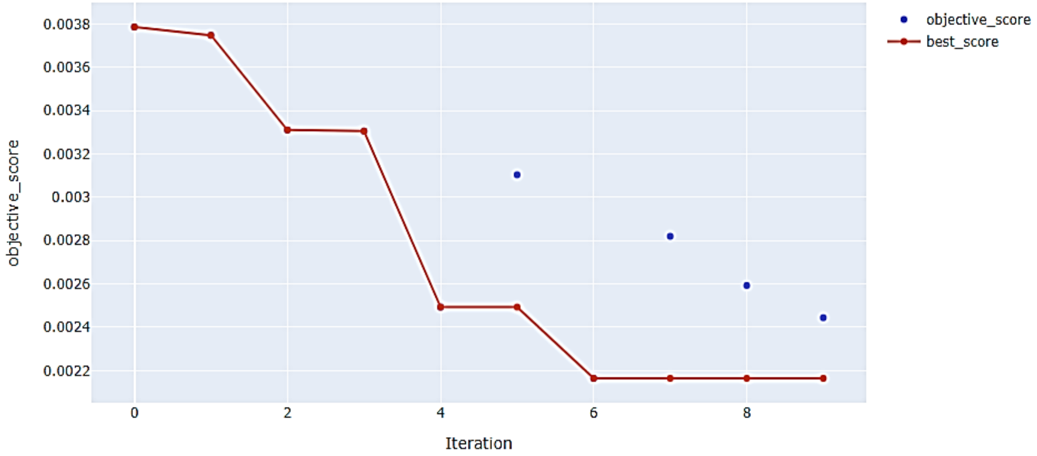

Figure 7. Optimization history plot

And

$$parentxi(t + 1)(t) + vi(t + 1) \qquad (4)$$

where vi(t) and xi(t) are the velocity and position of the ith particle at time t, w is the inertia weight, c1, and c2 are the acceleration coefficients, r1, and r2 are random numbers, pbesti is the personal best position of the ith particle, and gbest is the global best position of the swarm, as shown in Fig. 6. below.

In this work, PSO optimizes RF Classifier hyper parameters for binary classification. Optimization reduces validation dataset log-loss. An objective function, maximum of 10 iterations, population size of 20, and minimize parameter set to True initialize the PSO algorithm. PSO iteratively updates particle positions and velocities to obtain the best solution. Fig. 7 shows the method converges to the ideal solution or reaches 10 iterations.

To determine if 10 iterations are preferable, additional detail and clarity regarding what is being optimized and how the objective value varies with iterations. Several iterations of "bitterness" rely on various factors:

1. Objective function: The optimization objective function matters. It convex? Many local minima? These questions can impact the number of iterations needed to find a good solution.
2. Halting criteria: The stopping criterion affects iterations. One could end if the goal function change is below a threshold or after a given number of iterations.
3. Cost-quality tradeoff: Iterations frequently improve solutions, but they take more time and memory.

4. Results and discussion

This article discusses the honeypot system's model findings for malware detection. In a binary classification task categorize occurrences as malware or normal. To explore how hyper parameters affect classification performance, our models are tested with different sample sizes (60,000, 100,000, and 160, 000 examples) and K values (2, 3, 5, and 10).

4.1 Evaluation metrics

These metrics can help evaluate the performance of the proposed honeypot system using PSO and fuzzy logic algorithms for detecting malware attackers and ensuring data privacy and information security. Confusion metrics: A confusion matrix is a table that summarizes a classification algorithm's performance. It presents the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions for each class.

Accuracy: classifier accuracy. It is the percentage of correctly categorized cases, and the Precision: is the ratio of TP predictions to classifier positive predictions. It evaluates the classifier's malware detection and recall (TP rate): the ratio of TP forecasts to positive cases. It evaluates the classifier's malware detection, the F1-score: PRE and REC harmonic mean. It combines PRE and REC metrics. The Sensitivity: REC, the ratio of TP predictions to positive cases. It evaluates the classifier's malware detection and specificity: TN forecasts to total negative instances. It tests the classifier's usual case detection.

4.2 Libraries

This thesis listed libraries for ML preprocessing, visualization, optimization, and classification. These libraries built, optimized, and tested ML models for our honeypot system. Library preprocessing: Numpy: scientific python library. Pandas a data manipulation toolkit including DataFrame and Series data structures for cleaning, modifying, and analyzing data. It handles missing data, merges, restructures, and filters and Scikit-learn (sklearn

Visualization libraries:

Matplotlib: Popular python 2D charting library for static, interactive, and animated presentations. It provides an object-oriented chart API. Seaborn: Matplotlib-based statistical data visualization package. It draws appealing and useful statistical visualizations with a high-level interface.

Data optimization:

- Zoofs: Python module for metaheuristic feature selection optimization, including PSO.
- It reduces dataset characteristics, enhancing model performance and computational complexity.

Classifiers:

- Scikit-fuzzy: Python fuzzy logic library for fuzzy classifiers like fuzzy KNN. Fuzzy sets, membership functions, inference systems, and defuzzification are supported.

4.3 Tools

Table 1. Summarizes the results obtained from each model with different data shapes and K values. The performance metrics reported Include ACC, error rate, PRE, REC, F1-score, sensitivity, and specificity. The table below summarizes the results obtained from each model with different data shapes and K values. The performance metrics reported include ACC, error rate, PRE, REC, F1-score, sensitivity, and specificity.

For diverse data forms and K values, the models' metrics show relatively modest differences. Table 1 shows some trends: Model performance is stable from 60,000 to 180,000 instances, with just slight metrics changes. This implies that the FuzzyKNN model can handle bigger datasets without performance degradation. K value also has little effect on model performance. With minor differences, the models function well across K values. In a data set of 100,000 cases, K = 3 had somewhat greater ACC and specificity than K = 2, 3, and 10.

1. Random Forest (RF): RF constructs numerous decision trees at training time and outputs the mode of the classes of the individual trees [43]. This malware detection approach is accurate even when a lot of data is absent and can handle a lot of data according to [44].

2. Decision Tree (DT): DT is a flowchart-like structure with internal nodes representing features, branches representing decision rules, and leaf nodes representing outcomes. Root nodes are decision trees' highest nodes. It learns attribute-based partitioning. Recursive partitioning occurs. This decision-making flowchart helps. Visualization simplifies its interpretation by [45].

3. XGBoost—eXtreme Gradient Boosting. Gradient-boosting decision-tree-based ensemble Machine Learning algorithm. Artificial neural networks are the best at predicting unstructured data (pictures, text, etc.). Decision tree-based algorithms are best-in-class for small-to-medium structured/tabular data. Speed

Table 1. The FuzzyKNN classification report

Data shape	K	ACC	Error rate	PRE	REC	F1-score	Sensitivity	Specificity
60000	2	99.97%	0.0003	99.97%	99.97%	99.97%	99.97%	99.97%
	3	99.95%	0.0005	99.95%	99.95%	99.98%	99.97%	99.95%
	5	99.95%	0.0007	99.96%	99.95%	99.96%	99.97%	99.99%
	10	99.96%	0.0008	99.96%	99.96%	99.97%	99.97%	99.99%
100000	2	99.96%	0.0002	99.96%	99.97%	99.97%	99.97%	99.98%
	3	99.97%	0.0001	99.97%	99.97%	99.97%	99.97%	99.97%
	5	99.95%	0.00049	99.98%	99.97%	99.98%	99.97%	99.98%
	10	99.97%	0.00098	99.97%	99.97%	99.97%	99.96%	99.98%
160000	2	99.97%	0.00028	99.97%	99.97%	99.97%	99.98%	99.96%
	3	99.96%	0.00028	99.96%	99.96%	99.96%	99.97%	99.97%
	5	99.97%	0.00024	99.97%	99.97%	99.97%	99.96%	99.98%
	10	99.97%	0.00021	99.97%	99.97%	99.97%	99.96%	99.98%
200000	2	99.95%	0.00019	99.95%	99.97%	99.95%	99.97%	99.96%
	3	99.95%	0.00024	99.95%	99.95%	99.95%	99.96%	99.94%
	5	99.96%	0.00022	99.97%	99.97%	99.97%	99.96%	99.98%
	10	99.96%	0.00032	99.97%	99.96%	99.97%	99.95%	99.97%

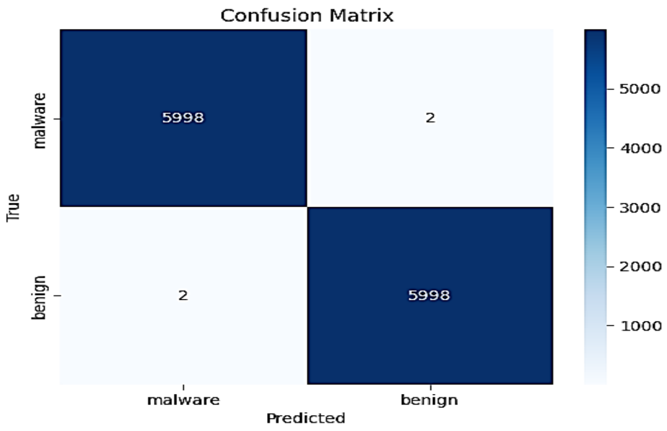

Figure. 8 FuzzyKNN with 60000 samples and K=2

and performance are XGBoost's major advantages according to [46]. Table 1: Compare Fuzzy_KNN with other Methods. in accuracy, recall, F1-score, and sensitivity and specificity. RF, DT, and XGBoost were lost to fuzzy KNN. DT had 99.97 percent accuracy and 0.0003 error rate, while RF had 99.98 percent accuracy and 0.00019 error rate. 98% accuracy and 0.02 error rate with XGBoost. The Fuzzy KNN algorithm performed best for our 5000-sample data collection. This comparison analysis shows the Fuzzy KNN method's reliability and accuracy for data categorization. Fuzzy KNN achieved 99.97 percent accuracy and 0.0003 error rate, it excelled.

4.4 Performance analysis

4.4.1. Sample size impact:

Across 60,000, 100,000, and 160,000 cases, the models performed well. The ACC ranged from 99.95% to 99.97% as the sample size grew. PRE, REC, and F1-score were 96% for most sample sizes and K values. The models detected malware and normal occurrences with great sensitivity and specificity. As the sample size rose, performance did not significantly fall. Performance indicators improved somewhat in certain situations, demonstrating that the FuzzyKNN model may scale well and maintain efficacy as the dataset expands according to [47].

4.4.2. Confusion matrices results

Confusion matrices show model performance for varied data forms and K values. The model predicts TP, TN, FP, and FN in each matrix. Fig. 8. depict K-value confusion matrices.

Dataset with 60000 samples (Fig. 8.)

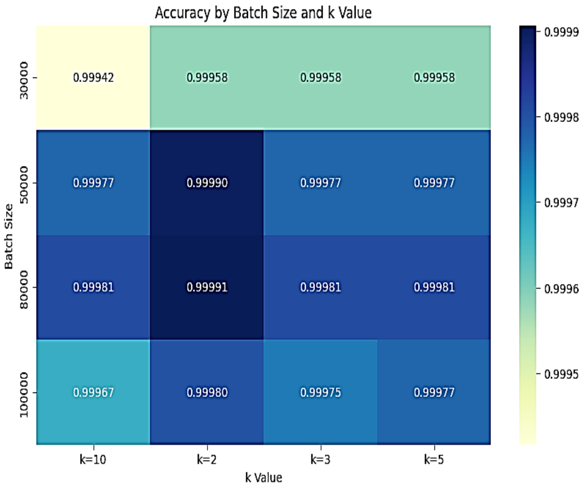

Figure. 9 Accuracy with different batch size and K value.

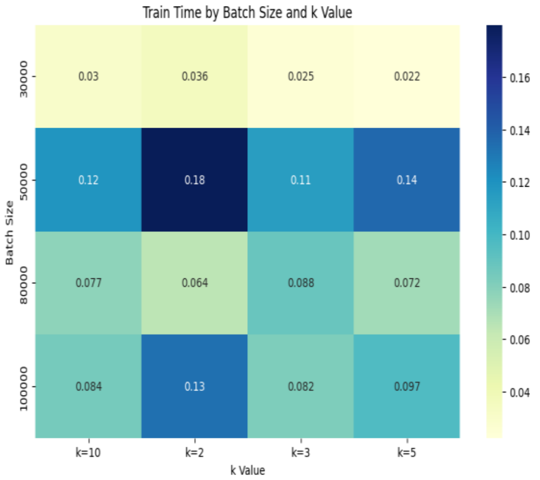

Figure. 10 Training time with different batch size and K value

4.4.3. Compared times and accuracy

As observed in Fig. 9, a clear trend is noticeable in terms of the impact of training samples and the choice of K in fuzzy K-nearest neighbors (Fuzzy_KNN) concerning both accuracy and training time. Primarily, an increase in the size of the training sample set from 60,000 to 100,000 is associated with an improvement in accuracy and an increment in training time. This trend is quite expected because larger datasets inherently offer a more comprehensive representation of the problem space, improving the accuracy of prediction models. However, they also necessitate more computational resources and time to process.

The analysis of Fig. 9 reveals insightful details regarding the impact of training samples and the choice of K in Fuzzy K-Nearest Neighbors (Fuzzy_KNN). When considering a dataset with 60,000 training samples, the accuracy consistently ranged between 0.9995 and 0.9996. Concurrently, the training time varied from approximately 0.022 to 0.035 across different values of K. However, with an increased number of training samples at 100,000, a slight improvement in accuracy was observed, ranging from 0.9997 to 0.9998. Nevertheless, this enhancement in accuracy was accompanied by a notable increase in training time, which expanded from 0.082 to 0.134. These specific values provide concrete evidence of the relationship between training sample size, accuracy, and training time in the context of Fuzzy_KNN, allowing for a more nuanced understanding of the trade-offs involved. The analysis depicted in Figure 10 offers intriguing insights regarding the impact of different values of K on accuracy and training time, considering various sample sizes. Surprisingly, higher values of K did not consistently yield superior accuracy or longer training times. For instance, when examining a dataset with 100,000 samples, the K=2 configuration resulted in the highest accuracy among all values of K. However, it also required the most time for training. On the other hand, the K=10 configuration exhibited slightly lower accuracy but necessitated less training time. These findings challenge the assumption that increasing K leads to improved accuracy or prolonged training times across different sample sizes. This observation underscores the importance of carefully selecting the optimal value of K based on the specific dataset characteristics and performance trade-offs to achieve the best possible balance between accuracy and training efficiency.

To summarize, the number of training samples and K in Fuzzy KNN interact complexly. More training samples provide more accurate but slower models. The optimum K seems problem-specific and does not follow the typical notion that bigger K values automatically result in more accurate models or longer training sessions.

4.4.4. Compared FuzzyKNN and fuzzy logic results

Let's examine these two models' accuracy, flexibility, and noise resistance:

1. Accuracy: Compared to fuzzy logic's 86% accuracy, Fuzzy KNN's accuracy ranges from 99.95% to 99.97%. Fuzzy KNN outperforms other models in this measure due to its better accuracy.

2. Flexibility: Fuzzy logic can handle ambiguity and vagueness, making it suitable for complicated real-world data. Fuzzy KNN, an instance-based learning algorithm, adapts well to fresh training data. It can adapt to new data patterns since it makes judgments depending on local data behavior. If mismanaged, this flexibility can cause overfitting.

3. Fuzzy KNN's "neighborhood" approach to classification makes it more robust to noisy input. This protects it from outliers and noisy instances. K—neighbors—is crucial. If K is too low, the model may be oversensitive to noise; if K is too high, it may miss important patterns. Fuzzy membership functions' design and implementation can minimize noise using fuzzy logic's capacity to handle ambiguity.

5. Discussion

As malware attacks get more complex, cybersecurity must change. Malware detection using practical swarm optimization (PSO) honeypots is being researched. Detailed comparison and debate are needed to evaluate this method. For years, cybersecurity relied on signature-based and heuristic malware detection. They seldom find zero-day malware. Swarm intelligence-inspired practical swarm optimization maximizes malware detection by simulating swarm dynamics. Practical swarm optimization's efficiency and scalability are key comparators to older approaches. In big networks, PSO algorithms optimize search space for faster, more accurate malware detection. Traditional approaches may fail to scale, delaying detection and response. Dynamic malware demands adaptive detection. PSO can quickly change detection strategies to changing threats by learning from swarm behavior. Traditional methods require manual updates and setup changes, making them less adaptive in real time. False positives in malware detection persist. PSO algorithms improve feature selection to reduce false positives and preserve accuracy. Traditional methods may miss threats or false alerts due to inaccurate and false positive rates. Resources must be used efficiently in cybersecurity. PSO optimization of detection saves computational resources and makes continuous monitoring more sustainable. Traditional methods may require a lot of computer power, increasing costs.

6. Conclusion

Honeypot technology improves cybersecurity, as proven by considerable study. Practical swarm optimization (PSO) and Fuzzy k-NN algorithms improve malware detection and mitigation in these systems to 99.97 percent. ACC, PREC, REC, F1-score, sensitivity, and specificity are robust binary classification metrics that distinguish malicious patterns from normal data, ensuring system reliability. Importantly, this study shows the system's exceptional performance across various sample sizes and K values, proving its scalability and versatility. In today's digital world, complex infrastructures are essential. Cyber threats are rising. This research also suggests future research. Honeypots are effective in detecting malware, but more study is needed to stay up with evolving threats. Future research should adapt models to new danger landscapes and various habitats. Automating K-nearest neighbors (KNN) hyper parameter K selection with cross-validation and machine learning is promising. Automation simplifies configuration and improves virus detection.

ML and DL models are promising for the future. Honeypot systems can detect complex cyber threats better by using these advanced methods.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Othman; methodology, Abu Alhija; software, Othman; validation, Alsharaiah, Abu Alhija, and Othman; formal analysis, Othman; investigation, Alsharaiah; resources, Abu Alhija; writing—original draft preparation, Othman; writing—review and editing, Alsharaiah and Abu Alhija; visualization, Alsharaiah; supervision, Abu Alhija.

Hebah Dar-Othman conceived and designed the study, performed experiments, analysed the data, and wrote the paper. Dr. Abu Alhija and Dr. Alsharaiah contributed to the literature review, methodology development, and critically reviewed the manuscript. All authors read and approved the final manuscript.

References

[1] A. V. Volkodaeva, A. V. Balanovskaya, and E. A. R. Nova, “Trends in information and communication technologies development in the context of economy digitalization”, *Digital Technologies in the New Socio-Economic Reality*, pp. 583-592, 2022.

[2] B. Bygstad, E. Øvrelid, S. Ludvigsen, and M. Dæhlen, “From dual digitalization to digital learning space: Exploring the digital transformation of higher education. Computers & Education”, Vol. 182, p. 104463, 2022.

[3] M. Attaran, “The impact of 5G on the evolution of intelligent automation and industry digitization”, *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-17, 2021.

[4] M. Knell, “The digital revolution and digitalized network society”, *Review of Evolutionary Political Economy*, Vol. 2, No. 1, pp. 9-25, 2021.

[5] M. Seete “The Digitisation of a Firm Process and Its Impact on Corporate Governance”, *Indian Journal of Corporate Governance*, Vol. 15, No. 2, pp. 280-294, 2022.

[6] A. Agrawal, “Corporate Governance and Technology in Modern Times: A Digital Transformation in Effective Governance”, *Indian JL & Legal Rsch*, Issue 4, Vol. 4, No. 1, 2022.

[7] S. Gangwar and V. Narang, “A Survey on Emerging Cyber Crimes and Their Impact Worldwide”, *Research Anthology on Combating Cyber-Aggression and Online Negativity*, pp. 1583-1595, 2022.

[8] A. A. Salih, S. R. Zeebaree, A. S. Abdulraheem, R. R. Zebari, M. A.Sadeeq, and O. M. Ahmed, “Evolution of mobile wireless communication to 5G revolution”, *Technology Reports of Kansai University*, Vol. 62, No. 5, pp. 2139-2151, 2020.

[9] B. Lutkevich, C. Clark, and M. Cobb, “Honeypot (Computing). TechTarget”, https://www.techtarget.com/searchsecurity/definition/honey-pot, last visited 30. Jun. 2023.

[10] M. S. Akhtar and T. Feng, “Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time”, *Symmetry*, Vol. 14, No. 11, p. 2308, 2022.

[11] C. Li, K. Mills, D. Niu, R. Zhu, H. Zhang, and H. Kinawi, “Android malware detection based on factorization machine”, *IEEE Access*, Vol. 7, pp. 184008-184019, 2019.

[12] M. S. Bhagat and S. Bajaj, “Malware detection using particle swarm optimization and fuzzy logic”, *SN Applied Sciences*, Vol. 3, No. 4, pp. 1-8, 2021.

[13] P. Yadav, N. Menon, V. Ravi, S. Vishvanathan, and T. D. Pham, “EfficientNet convolutional neural networks-based Android malware detection”, *Computers & Security*, Vol. 115, p. 102622, 2022.

[14] M. Baykara and R. D. SoftSwitch, “A centralized honeypot-based security approach using software-defined switching for secure management of VLAN networks”, *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 27, No. 5, pp. 3309-3325, 2019.

[15] R. Vishwakarma and A. K. Jain, “A honeypot with a machine learning-based detection framework for defending IoT-based botnet DDoS attacks”, In: *Proc. of 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019.

[16] O. Deepa and A. Senthilkumar, “Swarm intelligence from natural to artificial systems: Ant colony optimization”, *Networks (Graph-Hoc)*, Vol. 8, No. 1, pp. 9-17, 2016.

[17] I. Atacak, O. Çıtlak, and I. A. Doğru, “Application of interval type-2 fuzzy logic and type-1 fuzzy logic-based approaches to social networks for spam detection with combined feature”, *Peerj Computer Science*, Vol. 9, pp. 1-34, 2023.

[18] S. Phommixay, M. L. Doumbia, and D. Lupien S. Pierre, “Review the cost optimization of microgrids via particle swarm optimization”, *International Journal of Energy and Environmental Engineering*, Vol. 11, pp. 73-89, 2020.

[19] H. Wang and B. Wu, “SDN-based hybrid honeypot for attack capture. In 2019 IEEE 3rd Information Technology”, *Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1602-1606, 2019.

[20] L. Zhang, J. Wen, Y. Li, J. Chen, Y. Ye, Y. Fu, and W. Livingood, “A review of machine learning in building load prediction”, *Applied Energy*, Vol. 285, p. 116452, 2021.

[21] W. Ahmad, M. Arsalan, S. Nawaz, and F. Waqas, “Detection and Analysis of Active Attacks using Honeypot”, *International Journal of Computer Applications*, Vol. 184, No. 50, p. 27-31, 2023.

[22] K. Wang, M. Tong, D. Yang, and Y. Liu, “A web-based honeypot in IPv6 to enhance security”, *Information*, Vol. 11, No. 9, p. 440, 2020.

[23] W. Tian, X. Ji, W. Liu, G. Liu, R. Lin, J. Zhai, and Y. Dai, “Defense strategies against network attacks in cyber-physical systems with analysis cost constraint based on honeypot game model”, *Comput. Mater. Continua*, Vol. 60, No. 1, pp. 193-211, 2019.

[24] J. Hwang, J. Kim, S. Lee, and K. Kim, “Two-Stage Ransomware Detection Using Dynamic Analysis and Machine Learning Techniques”, *Wireless Personal Communications*, Vol. 112, No. 4, pp. 2597–2609, https://doi.org/10.1007/s11277-020-07166-9, 2020.

[25] S. Tiwari and A. Kumar, “Advances and bibliographic analysis of particle swarm optimization applications in electrical power system: Concepts and variants”, *Evolutionary Intelligence*, Vol. 16, No. 1, pp. 23–47, 2023.

[26] S. A. Aljawarneh, and M. A. A. Betar, “A new model for malware detection using honeypot and particle swarm optimization”, *Journal of*

Ambient Intelligence and Humanized Computing, Vol. 11, No. 3, pp. 1103-1113, 2020.

[27] C. Lin and H. Wang, "A Malware Detection Method Based on Honeypot and Particle Swarm Optimization", *International Journal of Online Engineering*, Vol. 17, No. 9, pp. 30-39, 2021.

[28] P. Wang, and H. D. Cruze, "Honeypots and knowledge for network defense", *Issues in Information Systems*, Vol. 22, No. 3, pp. 241-254, 2021.

[29] T. Kiran and P. Khandelwal, "Malware Detection using Honeypot and Fuzzy Logic", *International Journal of Advanced Science and Technology*, Vol. 30, No. 6, pp. 1666-1672, 2021.

[30] S. S. Gupta, and Dutta, "Malware Detection using Honeypot and Fuzzy Logic", *International Journal of Electrical, Computer, and Systems Engineering*, Vol. 15, No. 3, pp. 35-44, 2021.

[31] A. Jain and S. K. Sood, "Malware Detection using Honeypot and Fuzzy Logic", *Journal of Communication Engineering and Networks*, Vol. 3, No. 2, pp. 8-15, 2021.

[32] O. V. Arias, Á. J. González, P. G. García, C. M. Marin, and J. S. Cifuentes, "Applying fuzzy logic rules to predict computer attacks on honeynets", *Advanced Science Letters*, Vol. 25, No. 1, pp. 10-14, 2019.

[33] M. Dharshini and A. Tamilarasi, "Hybrid Malware Detection Technique Using Particle Swarm Optimization and Fuzzy Logic", *International Journal of Intelligent Systems and Applications*, Vol. 13, No. 5, pp. 59-69, 2021.

[34] J. Kennedy and R. Eberhart, "Particle swarm optimization", In: *Proc. of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.

[35] H. Ali, K. Batool, M. Yousaf, M. I. Satti, S. Naseer, S. Zahid, A. A. Gardezi, M. Shafiq, and J. G. Choi, "Security hardened and privacy preserved android malware detection using a fuzzy hash of reverse-engineered source code", *Security & Communication Networks*, 2022.

[36] I. Almomani, R. Qaddoura, M. Habib, S. Alsoghyer, A. A. Khayer, I. Aljarah, and H. Faris, "Android ransomware detection is based on a hybrid evolutionary approach in the context of highly imbalanced data", *IEEE Access*, Vol. 9, pp. 57674-57691, 2021.

[37] A. E. Ghamry, T. Gaber, K. K. Mohammed, and A. E. Hassanien, "Optimized and efficient image-based iot malware detection method", *Electronics*, Vol. 12, No. 3, p. 708, 2023.

[38] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proença, "Long short-term memory and fuzzy logic for anomaly detection and mitigation in a software-defined network environment", *IEEE Access*, Vol. 8, pp. 83765-83781, 2020.

[39] M. Abualhija, N. A. Shaf'i, N. M. Turab, and A. Hussein, "Encountering Social Engineering Activities with a Novel Honeypot Mechanism", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 13, No. 6, pp. 7056-7064, 2023.

[40] S. Abraham and I. C. Smith, "An overview of social engineering malware: Trends, tactics, and plications", *Technology in Society*, Vol. 32, No. 3, pp. 183-196, 2010.

[41] W. Ali, "Hybrid intelligent android malware detection using evolving support vector machine based on genetic algorithm and particle swarm optimization", *IJCSNS*, Vol. 19, No. 9, p. 15, 2019.

[42] Z. Avkurova, S. Gnatyuk, B. Abduraimova, S. Fedushko, Y. Syerov, and O. Trach, "Models for early web-attacks detection and intruders identification based on fuzzy logic", *Procedia Computer Science*, No. 198, pp. 694-699, 2022.

[43] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici "N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders", *IEEE Pervasive Computing*, Special Issue - Securing the IoT (July/Sep 2018).

[44] A. Mills, T. Spyridopoulos, and P. Legg, "Efficient and interpretable real-time malware detection using random forest", *In 2019 International Conference on Cyber Situational Awareness, Data Analytics, and Assessment (Cyber SA)*, pp. 1-8, 2019.

[45] L. Rokach and O. Maimon, "Decision trees. Data mining and knowledge discovery handbook", pp. 165-192, 2005.

[46] Ł. Podlodowski and M. Kozłowski, "Application of XGBoost to the cyber-security problem of detecting suspicious network traffic events", *In 2019 IEEE International Conference on Big Data*, pp. 5902-590, 2019.