International Journal of Artificial Intelligence
p-ISSN: 2407-7275, e-ISSN: 2686-3251

Original Research Paper

Design and Evaluation of a Fuzzy Logic Based Intrusion Detection System for Network Security

Ayomitope Isijola[1*], Emmanuel Afuadajo[1], Michael Asefon[2], Ufuoma Ogude[1], Jamiu Akande[3], Promise Joseph[1]

[1] *Department of Computer Sciences, University of Lagos. Lagos, Nigeria.*

[2] *Department of Computer Science, National Open University of Nigeria. Lagos, Nigeria.*

[3] *Cybersecurity and Artificial Intelligence, Birmingham City University. Birmingham, United Kingdom.*

Article History
Received:
26.07.2025

Revised:
19.08.2025

Accepted:
26.08.2025

***Corresponding Author:**
Ayomitope Isijola
Email
ayomitopeisijola@yahoo.com

Abstract: With the proliferation of networked systems, intrusion detection systems (IDS) have become vital in identifying and mitigating cyber threats and unauthorized access. Traditional IDS approaches, such as signature-based and anomaly-based methods, often struggle to detect novel attacks and tend to generate high false alarm rates. This study presents a robust, fuzzy logic-based IDS designed to detect network intrusions and assess their risk levels while minimizing false positives. The IDS classifies network intrusions by analyzing parameters such as source bytes, destination bytes, and packet rates, categorizing them into risk levels through defined fuzzy rules. Implemented in Python using libraries like scikit-fuzzy and pandas, the system utilizes the KDD Cup 99 dataset, a widely recognized IDS benchmark. Fuzzy membership functions and inference rules were defined for the primary input variables, enabling the system to infer intrusion likelihood. The IDS was tested using both two-variable and multi-variable input setups. It achieved a precision of 0.89, a recall of 0.85, and an F1-score of 0.87 in the multi-variable scenario. Results indicate that the fuzzy logic-based IDS achieves a balanced trade-off between detection accuracy and interpretability. It offers a transparent decision-making framework suitable for real-time applications due to its adaptability and potential for integration with live data streams. This research proposes future improvements by creating a foundation for hybrid intrusion detection systems (IDS) that integrate fuzzy logic and machine learning to enhance accuracy and interpretability. It recommends future research on adaptive fuzzy rules, real-time data processing, and explainable AI (XAI) to improve system flexibility, responsiveness, and transparency in cybersecurity applications.

Keywords: Fuzzy Logic, Intrusion Detection System, KDD Cup 99, Machine Learning, Network Security.

International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 81-100, December 2025. **DOI: 10.36079/lamintang.ijai-01202.870**

1. Introduction

Security problems are now a major problem in modern-day society, with concerns arising from various areas of human activity, specifically targeting network security. An Intrusion Detection System (IDS) has become crucial for network security as it aims to identify and stop malicious activities and attacks.

A foundational contribution in this field is provided by Javaheri et al. [1], who introduced a comprehensive classification of existing fuzzy logic-based methods for detecting Distributed Denial-of-Service (DDoS) assaults and broader network anomalies, establishing fuzzy logic as a practical paradigm for modeling the uncertain and imprecise nature of network traffic. Central to their analysis is the claim that traditional crisp decision boundaries used in rule-based and statistical IDS methods are inadequate in today's complex network environments, where attack patterns often exist in a gray area between normal and abnormal behavior. Fuzzy logic, with its capacity to represent gradual transitions and linguistic variables, presents a convincing alternative. The study highlights the theoretical benefit of fuzzy membership functions and fuzzy inference systems, which enable detailed risk classification, making them particularly effective for early detection of subtle anomalies and low-and-slow attacks.

Network perimeter defense relies on Intrusion Detection Systems (IDS) to locate and counter intrusions. IDS includes Host-based (HIDS), which analyzes local host data, and Network-based (NIDS), which examines network activity. IDS can be signature-based, locating intrusions by matching familiar patterns, or anomaly-based, recognizing deviations from normal behavior. Hybrid IDS combines both methods for enhanced detection. Network intrusion detection systems (NIDS) use fuzzy logic to manage uncertainty and improve decision-making by evaluating factors like packet size and connection rate. Fuzzy rules allow the system to adapt to changing threats, reduce false positives, and detect both identified and unidentified intrusions. This method enhances the effectiveness and adaptability of NIDS in dynamic network environments.

Jemili [2] provided critical insights into combining fuzzy logic and big data analytics for effective intrusion detection. The study suggests a new approach that merges fuzzy logic with advanced classification techniques designed for big data environments, offering a flexible model that adapts to the fuzziness and vagueness present in large network traffic data. It highlights fuzzy classification as a means to handle uncertain threat indicators within extensive datasets. Unlike binary detection systems that set strict decision boundaries, the fuzzy-based method manages uncertainty by assigning degrees of membership to different attack categories. This approach not only improves detection sensitivity for ambiguous or borderline cases but also enhances interpretability, which is essential in security operations where transparency in threat analysis is vital.

Network security is under constant threat from evolving attacks, making effective intrusion detection a critical concern. Traditional intrusion detection systems, both anomaly-based and signature-based, face significant limitations including an inability to detect novel and evasive attacks as well as high false alarm rates and scalability challenges.

The objectives of this study are to gather network traffic data, normalize it, and extract relevant features for input; to define fuzzy sets, membership functions, and key input variables such as source bytes and distance bytes; to develop fuzzy logic-based models by implementing the fuzzy inference system, applying rules, and performing fuzzification and inference; and to train and fine-tune the system using training data, evaluate its accuracy, and ensure that the fuzzy output is properly converted into a brittle result.

This research focuses on implementing a practical application of a Fuzzy Logic-based Intrusion Detection System (IDS). The Dataset - KDD Cup 99 is considered for this purpose. The methodology involves collecting and pre-processing the datasets, ensuring they include a broad scope of labeled network transmission data with both usual and attack patterns. The entire process develops a system to analyze complex, uncertain data by initial training on unclassified data and modifying it with classified datasets for improved rule-based detection. Performance will be evaluated using metrics like accuracy and detection rate, with attention mechanisms ensuring explainability and insights into the detection process. The system will overcome the constraints of traditional intrusion detection methods. It aims to effectively detect intrusions in network data and present the intrusion risk level while minimizing false alarms. Fuzzy logic will be deployed to handle imprecision in network data, defining linguistic variables and utilizing a rule-based fuzzy inference system. Public datasets will be utilized for the development and testing. It will combine misuse and anomaly detection techniques,

International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 81-100, December 2025. **DOI: 10.36079/lamintang.ijai-01202.870**

incorporating data mining for improved detection. The study aims to develop a more compatible, flexible, and precise intrusion detection system and advanced techniques. The study will not focus on hardware solutions or cryptographic methods. However, it is solely concentrating on software-based intrusion detection. While previous studies have used traditional signature-based and anomaly-based intrusion detection techniques, these approaches frequently struggle to identify new or evolving cyber threats, resulting in frequent false alarms and limited adaptability to changing network conditions. Although some research introduced machine learning or fuzzy logic techniques, many lacked real-time applicability, interpretable outputs, or relied on static rule sets that decrease responsiveness to new threats. Also, earlier fuzzy logic-based IDS implementations often used limited input variables or simple inference models, resulting in reduced scalability and accuracy in complex environments. This research addresses these issues by creating a multi-variable fuzzy logic-based IDS that enhances accuracy, adaptability, and interpretability, and also sets the stage for future hybrid models and real-time deployments.

2. Literature Review

In recent years, the integration of intelligent computing paradigms like fuzzy logic and evolutionary algorithms has gained considerable momentum in intrusion detection research. These methods focus on solving key problems in traditional IDS, such as high input data dimensionality, too many false positives, and poor ability to adapt to different attack types.

One notable contribution is the study by Reji et al. [3], who suggested a hybrid intrusion detection model that merges genetic algorithms (GA) with fuzzy logic inference to optimize feature selection and improve detection accuracy, particularly in identifying wormhole attacks, a type of network-layer threat common in wireless and ad hoc network environments.

Further theoretical development arises from the use of fuzzy support vector machines (SVMs), which tackle key challenges in classification, especially in environments with noisy, non-linearly separable data. Traditional SVMs, while powerful, are sensitive to outliers and assume a clear separation between normal and malicious traffic. However, by integrating fuzzy logic into the SVM framework, fuzzy SVM-based IDS systems can assign membership values to data points, thereby increasing robustness and flexibility in classifying ambiguous or borderline traffic patterns.

Expanding on these ideas, Subramani & Munuswamy [4] developed an intelligent IDS model that uses a Deep Fuzzy Convolutional Neural Network (DFCNN), combining the feature learning ability of convolutional networks with the uncertainty management of fuzzy logic systems. The model is especially useful in environments with diverse data flows and noisy or imprecise inputs, similar to the network datasets examined in this study. The fuzzy part of their model is key in managing soft boundaries between normal and malicious behavior, a principle that also supports the fuzzy inference approach used in this research to assess risk levels based on packet attributes such as source bytes, destination bytes, and packet rates.

This research addresses these gaps by:

1) Enhancing Accuracy & Interpretability
   Using fuzzy logic to classify intrusions into distinct risk levels, improving decision transparency, and reducing false positives.
2) Dynamic Multi-Variable Analysis
   Evaluating intrusion likelihood with both simplified and multi-variable configurations to achieve a more comprehensive assessment.
3) Real-Time Adaptability
   Implementing the IDS in Python with potential for integration into real-time network security frameworks.
4) Future Scalability
   Proposing dynamic rule tuning and machine learning integration to enhance IDS responsiveness to evolving threats.

By bridging these gaps, this study validates fuzzy logic as an extensible and practical approach to intrusion detection in complex networks.

International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 81-100, December 2025. **DOI: 10.36079/lamintang.ijai-01202.870**

2.1. The Development of Intrusion Detection Technology

The continuous evolution of digital threats has required corresponding advancements in Intrusion Detection Systems (IDS), especially to meet the safety needs of increasingly diverse and dynamic network environments.

A significant trend in IDS development is the combination of fuzzy logic with context-aware network models, especially in specialized environments like unattached sensor networks (WSNs) and mobile ad hoc networks (MANETs). For instance, Binthiya & Ravindran [5] designed a fuzzy logic-based IDS that uses a Hidden Markov Model (HMM) for WSNs. Their method observes node behavior, particularly energy levels, to identify black hole attacks, a common threat in these environments. By analyzing energy depletion patterns through fuzzy inference, the system successfully detects malicious nodes while maintaining Quality of Service (QoS) metrics such as packet delivery ratio, latency, and throughput.

Broadening the theoretical context, the comprehensive survey by Pinto et al. [6] presented an in-depth overview of the transformative role machine learning (ML) has played in IDS development. Their review synthesizes the current landscape of ML-driven IDS, emphasizing how data-driven approaches enable dynamic learning, automated pattern recognition, and improved adaptability to new and evolving threats.

Further expanding on the convergence of intelligent approaches, Gupta et al. [7] presented a current outline of the landscape of intrusion detection and prevention systems (IDPS), focusing on both the technological advancements and the architectural variety of existing solutions. The study categorizes IDPS into host-based and network-based systems, describing their detection techniques, which involve signature-based, anomaly-based, and hybrid methods. Their research highlights the increasing trend of incorporating machine learning and fuzzy logic into IDS to improve adaptability and intelligence. The survey emphasizes that hybrid models, which combine rule-based and learning-based systems, show greater promise in balancing accuracy with interpretability. Moreover, they recommend that future IDS adopt context-aware, scalable, and modular designs that can seamlessly integrate with live systems and modern infrastructure, aligning with the direction of this study's fuzzy logic-based IDS architecture.

In another advancement tailored for Internet of Things (IoT) and Routing Protocol for Low-Power and Lossy Networks (RPL), Kim et al. [8] introduced FLSec-RPL, a three-phase fuzzy logic-based IDS. Their framework starts with monitoring network behavior, then moves to fuzzy-based threat assessment, and finally implements dynamic response and mitigation, like separating doubtful nodes or modifying routing decisions. This shows how fuzzy logic not only helps in the detection stage but also acts as a control mechanism in network self-defense strategies.

2.2. Traditional Intrusion Detection Systems

Intrusion Detection Systems (IDS) have long been essential in protecting networks from unauthorized access and cyberattacks. However, as threats become more complex and subtle, traditional IDS methods frequently collapse, especially in locating zero-day assaults, adapting to new threat patterns, and processing vague or imprecise data inputs.

In this context, Hnamte & Hussain [9] advocate for shifting from static IDS models to deep learning-based frameworks, especially Deep Convolutional Neural Networks (DCNNs). Their research shows that DCNNs, when trained on large-scale network traffic datasets, can precisely identify complex malicious behavior patterns. Metrics such as detection accuracy and system efficiency significantly improved, particularly with GPU acceleration. The system's performance across four benchmark datasets demonstrated high detection rates, highlighting the effectiveness of deep learning in managing diverse intrusion scenarios in modern networks.

While deep learning overcomes some limitations of static models, its reliance on big amounts of classified data and heavy computation can impede real-time deployment and interpretability. As a result, researchers have turned to hybrid models that combine intelligence with flexibility. Alohali et al. [10], for example, introduced the IMFL-IDSCS (Intelligent Metaheuristic Fuzzy Logic-Based Intrusion Detection System for Cloud Safety). This architecture integrates fuzzy logic with metaheuristic optimization techniques to improve detection in cloud computing environments. Key components of their model include ECOA-FS (Enhanced Cuckoo Optimization Algorithm for Feature Selection) and ANFIS (Adaptive Neuro-Fuzzy Inference System), with JSSO (Jaya Seagull Swarm Optimization) used for dynamic tuning of membership functions. This framework demonstrates how

fuzzy logic's ability to handle uncertainty can be enhanced by optimization and adaptive learning to boost noticing accuracy and efficacy in robust cloud networks.

Diaz-Verdejo et al. [11] critically examined the detection performance of signature-based IDS, especially regarding modern web-based attacks. Their study reveals a growing gap between evolving attack methods and the static nature of signature-based defenses. Through rigorous testing, they demonstrate that these systems often fail to detect zero-day exploits, polymorphic malware, and increasingly complex injection attacks, highlighting their vulnerability in dynamic threat environments. A key theoretical contribution of their work shows how attack obfuscation techniques, such as encoding manipulation and parameter pollution, can effectively bypass signature detection mechanisms. The study also emphasizes that signature-based systems rely heavily on timely and comprehensive threat databases, a weakness that becomes even more significant in large-scale or distributed infrastructures where rapid response is crucial.

Traditional NIDS in Figure 1 are designed to identify malevolent activity by monitoring network traffic and comparing it with recognized attack signatures or by identifying abnormalities based on expected behavior. However, these often struggle with high false-positive rates as the network environments are dynamic and traffic patterns vary. To handle these, fuzzy logic provides a flexible decision-making mechanism that can better handle the vagueness present in real-time traffic data. The advantages of fuzzy-based NIDS include handling uncertainty in network data, adaptability to new attacks, and improved detection rates through fuzzy interpolation. The system gathers network traffic data, evaluates it against intrusion patterns using fuzzy rules, and processes input data with FIS models like Mamdani or TSK.

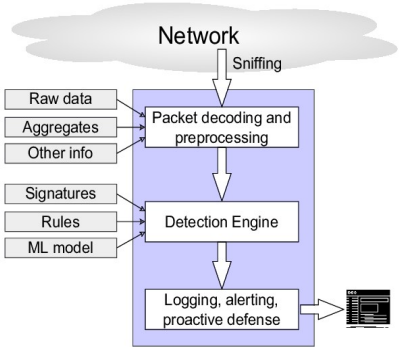

Figure 1. Traditional NIDS Architecture [12]

The contrast between rigid and adaptive methods is also explored by Kamboj et al. [13], who presented a detailed comparison of these two classical IDS approaches. Signature-based IDS relies on predefined designs to detect malicious activities, contributing top accuracy for identified dangers but failing to identify new or obfuscated attacks. In contrast, anomaly-based IDSs establish behavioral baselines and detect deviations, making them better suited for identifying unknown threats but more susceptible to false positives, often marking benign anomalies as malicious. Their comparison highlights several key concerns that have long challenged traditional IDS design. Notably, the static signature database limits responsiveness to emerging threats, while rigid anomaly thresholds in behavioral models often lack contextual flexibility. The authors believe that both methods, in their traditional forms, lack the interpretive depth and adaptability necessary to succeed in today's complex network environments, which are high-dimensional, noisy, and often ambiguous.

2.2.1 Signature-Based IDS

Signature-based Intrusion Detection Systems (IDS) are among the earliest and frequent methods for protecting computer networks from malicious threats. These systems work on misuse detection, where known attack patterns, also called signatures, are pre-encoded and stored in a database. The IDS constantly monitors network transmission or model activity and relates what it observes to a predefined dataset of attack signatures. When an alignment is detected, it marks it as an intrusion attempt and issues an alert. The core energy of signature-based IDS is its high accuracy in identifying

known dangers because it relies on exact pattern matching. This makes them especially effective in environments where the types of potential attacks are well understood and don't change often.

Kushal et al. [14], for example, offered valuable insights into this limitation by examining how modern systems try to reduce the rigidity of traditional SIDS through hybridization and automation. Their research introduced a self-adaptive IDS framework that merges signature-based detection with ensemble machine learning frameworks to create a more resilient and self-correcting system. This framework can dynamically retrain itself and adjust its behavior over time, marking a significant shift from traditional SIDS that rely solely on predefined rules and patterns.

With the evolution of signature-based IDS, a study examined multiple open-source signature-based IDS platforms and highlighted the benefits and trade-offs of diversifying detection engines within these systems. Their findings show that while diversity in rule sets and detection algorithms can improve overall coverage and reduce false negatives, it often adds management complexity and causes inconsistency in response behaviors. Specifically, the analysis pointed out that although a more varied ensemble of signature detectors can identify a wider range of threats, it also increases the challenge of maintaining compatibility and consistency across rule updates, especially when sourced from different communities or vendors [15].

2.2.2 Anomaly-Based IDS

Anomaly-based Intrusion Detection Systems (IDS) identify vicious activity by relating current action to a baseline of normal behavior. Any deviation from this norm is flagged as potentially suspicious. The deviations trigger an alarm in the anomaly detection engine. In an anomaly detection engine, any activity that deviates from the established or recognized model of behavior triggers events. How Anomaly-Based IDS Works:

1) Baseline Creation
   The system first undergoes a training phase where it learns what constitutes normal behavior for the network or system. This involves monitoring and analyzing typical user activities, network traffic, and system processes to create a normalized profile.
2) Anomaly Detection
   Once the baseline is fixed, the IDS continuously keeps an eye on real-time data in contrast to this profile. Any activity that substantially departs from the fixed baseline is flagged as an anomaly.
3) Alert Generation
   When a threat is detected, the system triggers warnings to inform executives of possible safety incidents, permitting for quick investigation and replies.

Anomaly detection techniques can be categorized into three kinds: Supervised, Unsupervised, and Semi-Supervised. Supervised methods use labeled data to classify instances as normal or abnormal, making them effective for known anomalies but less capable of detecting unknown threats. Unsupervised methods identify anomalies based on dataset characteristics without labels. Semi-supervised approaches combine labeled and unlabeled data for training and broader applicability. Fuzzy logic, data mining, and genetic algorithms significantly improve Intrusion Detection Systems (IDS). Fuzzy logic handles imprecision and vagueness by reasoning with varying degrees of truth, enabling the detection of complex patterns and variations in behavior. Data mining methods, such as association rules and recurrent episodes, uncover correlations and sequential patterns in audit data. Genetic algorithms optimize fuzzy membership functions and feature selection by encoding solutions as chromosomes. Together, these methods refine rule evaluation, pattern detection, and feature selection, enhancing IDS performance in identifying intrusions.

Building on these methods, Shenify et al. [16] proposed a novel fuzzy subspace clustering algorithm to detect anomalies in Internet of Things (IoT) environments. Their system uses fuzzy clustering to group data into overlapping subspaces, effectively managing the ambiguity and imprecision inherent in network traffic patterns. This approach allows the system to assign degrees of membership to multiple clusters instead of rigidly assigning a data point to a single one, thereby improving its sensitivity to anomalous behavior that may be near decision boundaries. The fuzzy subspace mechanism, in particular, helps the IDS identify relevant feature subsets dynamically, avoiding common issues like overfitting or underfitting that affect many high-dimensional anomaly detection models.

International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 81-100, December 2025. **DOI: 10.36079/lamintang.ijai-01202.870**

Similarly, Ramamoorthy & Karuppasamy [17] initiated a Unified Intrusion Detection Framework (UIDF) created to improve intrusion prediction in wireless sensor networks (WSNs). The UIDF functions through a layered architecture that combines data pre-processing with behavioral modeling. Notably, it employs a predictive analytics approach by analyzing historical traffic patterns and forecasting future states, enabling it to identify deviations as potential intrusions. This approach advances beyond static thresholding or pattern-matching by incorporating temporal dynamics and contextual understanding into the anomaly detection process.

2.3. Fuzzy Rule-Based System

Fuzzy Rule-Based Systems (FRBS) are systems that use fuzzy logic to make decisions based on fuzzy rules, allowing for reasoning in uncertain environments. These systems consist of key elements: fuzzification, rule base, inference engine, and defuzzification. Fuzzification changes real-world inputs into fuzzy sets, while the rule base defines connections between input and output fuzzy sets through if-then rules. The inference engine determines applicable rules based on input data and combines results to form a fuzzy output using operations like min-max or product-sum. Defuzzification then changes the fuzzy output into a clear value for decision-making. An example of an FRBS in practice is a fuzzy control model for an air conditioner, where inputs like ambient temperature and humidity level are fuzzified into fuzzy sets and used to trigger specific rules for setting the cooling level. Overall, FRBS enables nuanced decision-making in scenarios where data is imprecise, making them essential components in various applications such as control systems and decision-making systems. Defuzzification is the task of interpreting a fuzzy set portraying the Cooling Level into a crisp value. Fuzzy Rule-Based Systems (FRBS) are utilized in Intrusion Detection Systems (IDS) for observing network tasks and detecting potential threats. Fuzzification in IDS involves converting data like packet size and traffic volume into fuzzy values. The Rule Base consists of rules to locate normal and abnormal network behavior, while the Inference Engine processes these rules to determine the degree of suspicion for different types of attacks. Defuzzification in IDS converts fuzzy outputs into threat scores between 0 and 1.

Fuzzy rules are essential components of fuzzy logic systems, allowing for the inference of outputs based on input variables. These rules manage uncertain or imprecise data by utilizing fuzzy sets. The rule structure includes an antecedent (IF) specifying input conditions and a consequent (THEN) determining the outcome when conditions are met to a certain degree. Each condition links an input variable to a fuzzy set, evaluating its truth degree to produce a consequence. In a network intrusion detection system (NIDS), fuzzy rules analyze parameters like packet size, connection rate, and CPU usage to assess intrusion risks. If the packet size is large, the connection rate is high, and CPU utilization is high, then the system infers a high intrusion risk. Fuzzy logic operates through fuzzification, transforming precise input values into fuzzy sets to handle vague data effectively.

Recent studies highlight the growing relevance of fuzzy logic in IDS, particularly in constrained or decentralized environments like IoT and Wireless Body Area Networks (WBANs). Muniyandy et al. [18] proposed a Fuzzy Intrusion Detection Technique integrated with Zero-Knowledge Validation, specifically designed for Internet of Things (IoT) networks. Their model uses fuzzy inference systems to evaluate network traffic anomalies by classifying input attributes such as packet type, source ID, and message frequency into fuzzy sets, then applies a rule base to determine the risk of intrusion. What sets their work apart is the incorporation of zero-knowledge protocols, enabling lightweight, secure device authentication without revealing sensitive data, thus further strengthening system integrity.

Bengag et al. [19] developed a fuzzy inference system (FIS) that assesses signal strength, energy levels, and transmission success rates, which are vulnerable to manipulation during jamming attacks. These input variables were converted into language classifications, like low, moderate, and high, and a rule base was created to determine the likelihood of an intrusion. Their fuzzy system did not depend on historical data or attack signatures; instead, it made real-time inferences based on dynamic behavioral thresholds, significantly decreasing false positives and allowing for proactive responses to subtle, evolving threats.

In a related study, Bengag et al. [20] addressed the limitations of conventional IDS techniques in WBANs, especially in detecting low-intensity or new attack vectors. Recognizing the flexibility of fuzzy logic, they suggest an IDS framework that uses fuzzy inference to analyze key parameters such as signal strength changes, energy use, and transmission delay. Their main contribution is showing

how fuzzy rule-based models, created with expert knowledge, can evaluate risk levels dynamically, even in environments with limited computational resources and bandwidth.

2.4. Hybrid Fuzzy Approaches

Several researchers have explored hybrid methods that join fuzzy logic with other methods. The integration of optimization algorithms with fuzzy rule-based reasoning introduces adaptability and intelligence to systems traditionally constrained by static rule sets, offering improved detection capabilities and reduced false alarms.

A typical example is the work by Jain et al. [21], who introduced an innovative hybrid IDS model that joins the strengths of fuzzy logic and deep neural networks (DNNs), trained using a bio-inspired optimization method called the Honey Badger Algorithm (HBA). Central to their approach is a fuzzy-deep neural network architecture, where fuzzy systems preprocess input features to capture vague and overlapping data patterns, which are common in real-world network traffic. These preprocessed features are then fed into a deep learning model capable of uncovering complex, non-linear relationships. This hybrid approach ensures both semantic interpretability through fuzzy rules and high classification accuracy from the deep neural network.

Building on this integration, Raj & Pani [22] posed a new hybrid intrusion detection system that joins a Deep Residual Fuzzy Network (DRFN) with the White Shark-Dwarf Mongoose Optimization (WSDMO) algorithm. This model shows the merging of fuzzy inference systems with deep learning-based feature extraction and nature-inspired optimization methods. The DRFN improves feature extraction by using residual learning, which helps maintain important network behavior features across layers. Fuzzy logic is then used to handle the uncertainty in network traffic patterns, providing detailed threat level classification. What sets this approach apart is the WSDMO algorithm, a biologically inspired metaheuristic that adaptively adjusts fuzzy rule parameters and neural network weights to boost detection accuracy and convergence speed.

Also, Sekar et al. [23] introduced a hybrid Bayesian Decision and Fuzzy Logic System for both intrusion detection and avoidance, highlighting the potential of combining statistical inference with fuzzy reasoning in cybersecurity. Their model uses Bayesian decision theory to compute the posterior probability of an intrusion event based on observed network features, supporting data-driven decisions under uncertainty. This probabilistic output is then interpreted within a fuzzy logic framework, which improves the system's strength to evaluate threat severity and adjust to imprecise or overlapping input patterns, such as variations in packet rates, source or destination bytes, or other anomaly indicators.

For resource-constrained environments like IoT networks, Bamidele et al. [24] presented a robust model that combines multi-level Random Forest (RF) classification with a Fuzzy Inference System (FIS), specifically designed for Internet of Things (IoT) networks, which are often restricted in resources and susceptible to numerous security threats. In their framework, the Random Forest model acts as an initial classifier, sorting network traffic into various risk levels based on learned patterns from extensive network data. These results are then fed into a Fuzzy Inference System, which further refines decision-making by applying fuzzy rules to assess the likelihood of intrusion. This two-layer system provides a layered view of threats, with the Random Forest offering generalization and learning capabilities, while fuzzy logic handles uncertain or borderline cases more delicately.

2.5. Addressing False Alarms

One of the main problems in IDS is reducing false alarms while maintaining high detection rates. To mitigate this, recent research has explored hybrid methods that integrate advanced optimization techniques with fuzzy logic to enhance classification precision without sacrificing sensitivity.

A significant contribution is the work by Ling et al. [25], who introduced an IDS framework that combines rough-fuzzy set theory with a Parallel Quantum Genetic Algorithm (PQGA) to refine detection boundaries and greatly reduce false alarms. In this approach, rough set theory preprocesses the data by identifying and removing redundant or noisy attributes, thereby focusing the system on the most relevant features for intrusion classification. This dimensionality reduction step not only simplifies processing but also prevents the system from reacting to harmless anomalies.

False alarms in Intrusion Detection Systems (IDS) overwhelm administrators and reduce effectiveness. Strategies to mitigate this include improved data preprocessing (noise reduction, feature selection), anomaly detection calibration (threshold tuning, behavioral profiling), hybrid detection methods (combining signature-based and anomaly-based methods, using fuzzy logic), AI (supervised

learning, adaptive learning), and context-aware intrusion detection. These methods enhance IDS performance, ensuring accurate threat detection and preventing costly misjudgments.

Contextual analysis, cross-referencing data, hybrid fuzzy approaches (e.g., Neuro-Fuzzy, Fuzzy Genetic Algorithms), multi-layered designs, human-in-the-loop validation, Bayesian filtering, and temporal/sequential analysis are all techniques that improve Intrusion Detection Systems (IDS) by decreasing false alarms and improving accuracy in identifying real threats. These methods achieve this by better distinguishing malicious activity from normal network anomalies.

2.6. Intrusion Detection in Cloud Security

Cloud computing offers easy, anytime access to computing resources through virtualization techniques. Gartner defines it as scalable IT services delivered over the Internet. Cloud infrastructure uses virtualization and standard Internet protocols but faces security vulnerabilities. Firewalls protect system access points but struggle with insider attacks. Efficient intrusion detection and prevention mechanisms are needed in cloud environments to counter insider threats. The effectiveness of these systems depends on factors like technique, positioning, and configuration within the network. Addressing this, Ghazal et al. [26] offered a fuzzy logic-based framework for detecting security threats in cloud infrastructures, highlighting the system's capability to reason under uncertainty and manage ambiguous or partial network information. Their approach uses fuzzy inference mechanisms to assess the risk of potential security breaches by analyzing fluctuating cloud metrics such as resource usage anomalies, traffic irregularities, and authentication patterns. A key strength of the model is its application-focused design, which adjusts detection strategies based on the type of cloud service in use (e.g., IaaS, PaaS, or SaaS), emphasizing the need for context-aware IDS in complex, virtualized environments. Their system was tested through simulated attack scenarios, showing that fuzzy logic can effectively identify unusual patterns while maintaining computational efficiency, an essential requirement for real-time cloud security applications.

Traditional IDS mechanisms, often designed for static network environments, struggle to remain effective in cloud settings due to heterogeneous architectures, elastic resource allocation, and rapid changes in traffic. Nevertheless, the shift to the cloud platform comes with security and operational difficulties. Security concerns arise due to the increasing trend of cloud-based data storage, with vulnerabilities in technologies for virtual machines and hypervisors being exploited by attackers for data exfiltration and various types of assaults. The open and dispersed nature of the cloud structure yields a prime target for invaders, leading to conventional network and cloud-specific assaults that jeopardize cloud users. Intrusion detection systems are essential in cloud security by giving an additional layer of defense opposed to known and unknown assaults. Intrusions, whether initiated by aggressors, legal users seeking unauthorized privileges, or users misusing their access rights, can be identified through intrusion detection techniques. An Intrusion Detection System tracks network activity and notifies network managers of potential intrusion events, whether hardware, software, or a blend of both. However, false positives and false negatives impact the usefulness of the Intrusion Detection System, highlighting the need for continuous advancement in detection efficiency and the employment of feature selection methods to enhance system accuracy.

In a related study, Alrayes et al. [27] introduced an optimal fuzzy logic-enabled intrusion detection model (IDS) especially created for IoT-cloud surroundings, which are marked by extensive device connectivity, variable data flows, and low-latency needs. Their approach combines fuzzy logic with optimization techniques to enhance detection accuracy and reply time in layered infrastructures. The system models various input features, such as packet delay, energy consumption, and data transmission behavior, using fuzzy membership functions to capture the ambiguity and unpredictability inherent in IoT-cloud traffic. A fuzzy inference engine is then used to evaluate the likelihood of intrusion based on these uncertain parameters. The researchers also incorporated an optimization algorithm, likely genetic or swarm-based, to adjust fuzzy rule sets and threshold values, ensuring the system’s adaptability to changing network conditions.

Also, Khordadpour & Ahmadi [28] addressed the problem of detecting attacks in cloud ecosystems by developing FIDS (Fuzzy Intrusion Detection System), a fuzzy logic-driven approach tailored for the simultaneous identification of DoS and DDoS assaults in cloud computing environments. Their study is rooted in the notion that traditional detection systems, particularly those relying on binary classifications and fixed thresholds, often fall short in dealing with the subtle, evolving characteristics of modern network traffic. Instead, FIDS leverages fuzzy logic’s inherent ability to manage

International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 81-100, December 2025. **DOI: 10.36079/lamintang.ijai-01202.870**

imprecision and uncertainty to detect intrusions with greater nuance and flexibility. The proposed system models network behavior using fuzzy linguistic variables derived from real-time input features such as request rates, source IP entropy, and response anomalies. By applying a fuzzy inference mechanism, FIDS evaluates the likelihood and severity of attack patterns based on predefined fuzzy rules that emulate expert reasoning. One of the central contributions of the study lies in its ability to simultaneously assess multiple forms of denial-of-service behavior, thereby reducing detection latency and avoiding the need for multiple independent systems.

2.6.1. Real-Time Detection and Scalability

With the rapid growth of cloud-based and networked systems, ensuring the security of virtualized environments has become increasingly important. Cloud infrastructures, defined by their distributed, elastic, and on-demand features, are especially vulnerable to cyber threats due to their high traffic levels and dynamic resource allocation. In this context, Intrusion Detection Systems (IDS) are crucial in safeguarding data integrity, availability, and confidentiality by identifying and preventing unauthorized access and malicious activities.

Traditional IDS methods, like signature-based and anomaly-based techniques, often fall short in cloud environments since they cannot identify new or zero-day assaults and tend to generate many false alarms. These challenges highlight the demand for more flexible, scalable, and smart detection systems. Recent developments in deep learning show significant promise in this area. For instance, Hnamte & Hussain [29] presented a Deep Convolutional Neural Network (DCNN)-based IDS framework that achieved excellent detection accuracy, ranging from 99.79% to 100% across multiple datasets. While these methods deliver strong performance, they frequently function as “black boxes”, lacking transparency and requiring extensive computational power, factors that can limit their use in real-time cloud environments.

2.6.2. Specific Attack Detection

As security attacks become more polished, intrusion detection systems (IDS) must evolve beyond basic anomaly detection to accurately identify specific types of attacks in real time, especially within cloud-based networks where attack surfaces are broad and constantly changing. Among the emerging solutions, fuzzy logic has made an impression for its strength to handle uncertain and imprecise information, a common characteristic of network traffic during attacks.

Almseidin et al. [30] demonstrated this potential through a fuzzy logic inference system designed for Distributed Denial of Service (DDoS) assault detection. By analyzing traffic anomalies utilizing defined fuzzy rules, the system achieved a correct positive rate of 91.1% and an incorrect positive rate of just 0.006%, effectively showcasing the applicability of fuzzy logic in detecting complex, large-scale attack patterns with high precision and minimal false alarms.

2.7. Limitations of Conventional Approaches

Conventional intrusion detection systems (IDS) countenance several extents such that hinders their efficacy in modern cybersecurity environments. Below are some of the key challenges associated with these traditional approaches:

- High False Positive Rates
  Conventional IDS often yields high false positives, overwhelming security teams and causing alert fatigue, which may lead to missing genuine threats.
- Short Detection Rates for Zero-Day Assaults
  Traditional IDSs use predefined signatures to detect identified threats, resulting in inefficiency opposed to zero-day attacks and emerging threats.
- Limited Visibility and Context
  IDS focused on traffic monitoring might miss threats without noticeable network activity. Encrypted traffic hinders effective content analysis, allowing undetected malicious activities.
- Evolving Threat Landscape
  Cybersecurity is ever-changing, and attackers use new techniques to evade detection. Outdated IDS may not keep up, needing constant updates to combat new threats.
- Insider Threat Detection Challenges
  Traditional IDS focuses on external threats and may miss insider threats. User behavior analytics is needed to monitor internal activities.

International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 81-100, December 2025. **DOI: 10.36079/lamintang.ijai-01202.870**

3. Methodology

The methodology of a Fuzzy-Based IDS integrates fuzzy logic principles into two key operational phases: rule-generation and detection. These phases collectively enhance the model's strength to manage and interpret unpredictable and inaccurate network data, resulting in more accurate and robust intrusion detection.

The system operates in two modes: Rule-Generation and Detection. In Rule-Generation mode, we used historical network data to create a model of network behavior. The data in this phase is collected and converted into fuzzy values, allowing for the identification of patterns using an algorithm for fuzzy data mining. We then generated Fuzzy rules and refined the rules to distinguish normal activities from malicious activities. In Detection mode, real-time network data is monitored and converted into fuzzy values.

We then applied an inference engine to fuzzy rules to detect anomalies, with defuzzification translating these findings into actionable decisions. The system provides an output indicating the detection of an intrusion level, facilitating timely responses to potential threats. Both modes leverage fuzzy logic to handle uncertainties and adapt to changing network behaviors, ensuring effective network security measures.

3.1. Data Preprocessing Module

The preprocessing phase is the first part of our proposed system. Training data is received as a system log by this module, which then divides it into two classes: normal and abnormal data. The dataset in consideration is KDD Cup 99, which contains various types of assaults: Denial of Service, Remote to Local, User to Root, Probe, and ordinary behavior data. This data is utilized to create rules. The preprocessing module, which involves data collection, consists of two smaller modules: the data miner and the attribute selector.

- Data miner
  The data miner module is responsible for extracting meaningful patterns and relationships from the preprocessed network data. The data miner will integrate the property of the FP-Growth algorithm to mine the most common thing in each attribute. A data miner efficiently mines frequent itemsets in large datasets by providing enough support. Both classes' frequent items are identified, which include normal and abnormal classes.
- Attribute selecto
  This phase is responsible for identifying and selecting the most relevant attributes from the preprocessed network data, since none of the characteristics are helpful for detection. We deployed the deviation method to identify the suitable attributes. We used the selected attributes to produce rules for the proposed system. We then calculate the range of the effective attributes' deviations for both normal and aberrant data to derive the point of intersection. The point of intersection is used to create the IF-THEN rules.

1) Input Variables

The input variables are the measurable parameters that the fuzzy system will analyze to determine the likelihood of an intrusion. In the context of network traffic, we considered three input variables, namely:

- Source Packet Rate
  This is the number of data packets transmitted over a network within a specific time frame, often measured in packets per second (PPS).
- Connection Duration
  This is the period a connection is established between a client and a server, measured in seconds.
- Source Bytes
  This is the sum of bytes transferred from an origin node or device in a network.
- Distance Bytes
  This is the amount of data transferred over a given "distance" or between nodes in different network segments, but this would be a unique or specialized usage.
- Distance Packet Rate
  This is the rate of packet distribution between two network nodes or segments over a specific distance, usually calculated in packets per second (PPS).

International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 81-100, December 2025. **DOI: 10.36079/lamintang.ijai-01202.870**

2) Data Normalization

This step is crucial during the data preprocessing phase. It entails normalizing the values to a consistent range in the interval [0, 1] to make sure all features have equal impact on the analysis. This step is crucial in avoiding attributes with wider ranges from overpowering calculations, particularly in algorithms that are scale-sensitive.

3.2. Fuzzy Rule Generator

The fuzzy rules come from the precise rules acquired through the aforementioned procedures. The specific rules have just a single categorization label in the THEN section. The fuzzy rules involve variables in linguistics exclusively, so the membership function assigns a number between 0 and 1 to every element in a specified input space. This value shows how much the element is part of a fuzzy set and is used to make numerical values fuzzy in rules. In our analysis, we employed triangular and trapezoidal membership functions. For instance, when attribute1 is A1, the data is considered abnormal, whereas when attribute1 is A2, the data is considered normal. This means that both A1 and A2 are linguistic variables. Next, the fuzzy inference system Rulebase receives these fuzzy rules.

3.3. Fuzzy Inference System

This stage explains the fuzzy logic method for determining the appropriate category of the test data input. It requires configuring the system to utilize specific fuzzy sets and membership functions to analyze the input test data and generate an outcome. The Mamdani fuzzy inference system is employed in this system. The steps include setting input and output variables, establishing membership functions, using fuzzy rules, carrying out fuzzification and inference, and ultimately defuzzification for a precise output.

3.4. Determining Behavior for a Test Input

During the testing module, the Mamdani fuzzy inference system receives a set of test data from the testing data. It evaluates the input variable against the membership functions to determine the membership value of each linguistic label. The decision-making unit compares the inputs with the Rulebase after receiving the output from the fuzzification interface. The knowledge base's result is then input into the defuzzification boundary, which transforms fuzzy outcomes into a clear value. If the sharp value falls within the specified range for each bearing, the data will be categorized as either normal or abnormal.

3.5. Algorithm

Input: Performance record documented by the computer system
Output: Intrusion level
Assumptions:

- Training data is classed under the various attacks: Normal, Probe, DOS, RLA, and URA.
- Testing data contains data classed under the various attacks: Normal, Probe, DOS, RLA, and URA.

Step 1 *An attribute selector and data miner are included in a preprocessing module that receives training data as input.*

- *Data miners efficiently mine regular itemsets in large datasets identified for both classes, which include normal and abnormal classes.*
- *The attribute selector identifies and selects the most relevant qualities that are appropriate for rule-generation.*
- *With the use of the intersection point of the efficient qualities, the IF-THEN rules are created.*

Step 2 *The rules derived from the preprocessing module are supplied to the module that generates fuzzy rules.*

Step 3 *Use the selected attributes and generate the IF-THEN rules for the fuzzy inference system.*

- *Outline triangular and trapezoidal membership functions for each attribute.*

International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 81-100, December 2025. **DOI: 10.36079/lamintang.ijai-01202.870**

- *Design fuzzy rules based on the selected attributes and their membership functions.*

Step 4 *Fuzzifying the inputs by utilizing the input membership functions,*
Step 5 *Matching up the fuzzified inputs based on the fuzzy rules to design a rule strength,*
Step 6 *Determine the result of the rule by matching up the rule strength and the output membership function,*
Step 7 *Combine the results to derive an output distribution,*
Step 8 *Defuzzifying the output distribution.*

4. Finding and Discussion

This part presents the outcomes of the experiments and the functioning of the suggested system. The system proposed is executed in Python and its performance is assessed making use of precision, recall, and F-measure.

For the exploratory assessment, we utilized the KDD Cup 99 datasets, commonly employed to evaluate intrusion detection system performance. Evaluating the effectiveness of the recommended system on the KDD Cup 99 dataset is challenging due to its large scale. We used a 10% subset of the KDD Cup 99 dataset for both training and testing purposes.

4.1. Programming Language

Python is a high-level, interpreted, and object-oriented scripting language. Python is meant to be very legible. It has fewer syntactic features than other languages and uses English terminology instead of punctuation.

- Python is interpreted.
- It is an object-oriented programming language.
- It is easy to comprehend.

Python is ideal for developing fuzzy logic-based intrusion detection systems (IDS) because of its extensive library ecosystem (e.g., scikit-fuzzy, pandas, scikit-learn), user-friendly experimentation with tools like Jupyter Notebook, powerful visualization options (Matplotlib, Seaborn), real-time integration capabilities (e.g., Apache Kafka), cross-platform scalability, active community support, and smooth integration with machine learning models for enhanced accuracy and robustness.

4.2. Implementation Requirement

The model requirements are separated into two primary sections: software requirements and hardware requirements. The data's credibility is a key aspect of the process for this research.

1) Software Requirements: Python 3.9.0, JupyterLab
2) Hardware Requirements: HP Intel(R) Core i5, 2.00GHz Processor 8GB RAM, 64-bit OS
3) Dataset Requirement:
   - Source: Kaggle.com
   - Type of Dataset: KDD Cup 99

4.3. Experimentation

4.3.1. Importing Libraries

1) Scikit-Fuzzy (Scikit-Fuzzy)

Purpose: The scikit-fuzzy library is essential for fuzzy logic systems in Python. We utilized Scikit-fuzzy to construct fuzzy logic systems that are beneficial for decision-making, control systems, and machine learning due to their ability to manage uncertain or imprecise information.

Key Features:

- Fuzzy Variable Definition
  It allows for creating fuzzy variables (antecedents and consequents) with defined membership functions (e.g., low, medium, high), which can handle uncertainty in data.
- Membership Functions
  It provides a variety of membership functions, like triangular, trapezoidal, Gaussian, etc., to model linguistic terms (like "low speed" or "high temperature").
- Fuzzy Rule Systems

It enables the creation of fuzzy rules to explain how a system's inputs and outputs compare to one another. For example, "If the temperature is high and the humidity is low, then the risk is medium."

- Fuzzy Inference and Control
  It facilitates fuzzy inference, which evaluates the rules and computes the output based on given inputs, and control systems for applications like automated decision-making and control.
- Defuzzification
  It provides tools for defuzzification, which converts fuzzy results back to a precise output (e.g., converting a fuzzy "high" risk into a numeric value).
- Usage Example
  To create a simple fuzzy control system.

Figure 2 presents some packages that can be useful for training the model.

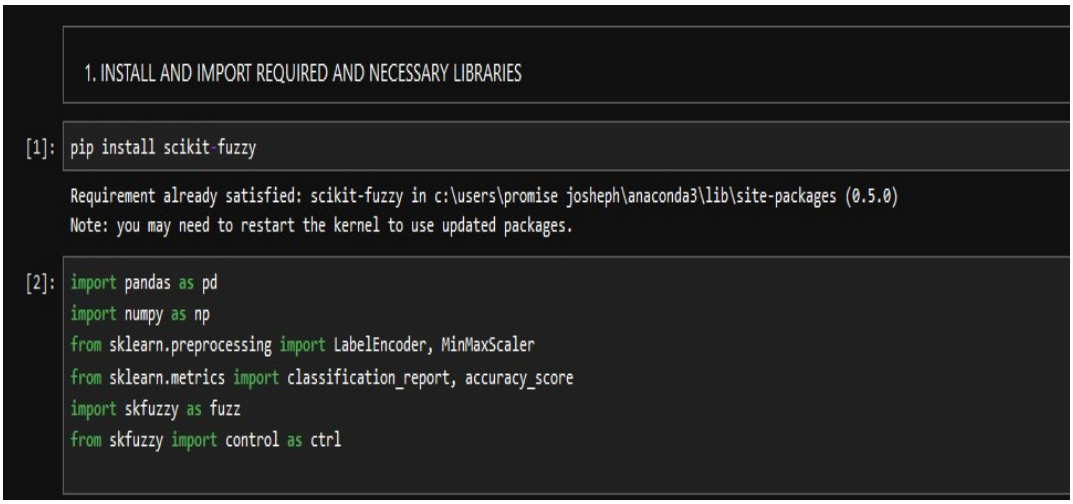

Figure 2. Importing the Necessary Python Packages

2) Pandas

Purpose: Pandas operate as a robust tool for data analysis and manipulation. It is commonly utilized for managing and manipulating structured data, especially in DataFrame form.
Key Features:

- DataFrames and Series
  This allows for easy manipulation of tabular data through DataFrames (2D) and Series (1D).
- Data Cleaning
  This provides platforms for managing misplaced data, duplicates, and other data-cleaning operations.
- Data Analysis
  This facilitates data aggregation, filtering, grouping, merging, and pivoting.
- I/O Operations
  Can read/write from/to vast file formats like CSV, Excel, SQL databases, etc.
- Usage Example
  Pandas simplifies complex data analysis and transformation tasks, making it an important tool for data scientists and analysts.

3) Numpy

Purpose: Numpy is the main library for performing numerical and scientific computations in Python. It offers assistance for big, multi-dimensional arrays and matrices, as well as mathematical functions for working on them.

International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 81-100, December 2025. **DOI: 10.36079/lamintang.ijai-01202.870**

Key Features:

- Multi-dimensional Arrays
  This supports efficient manipulation of large datasets through arrays.
- Mathematical Operations
  This offers functions for linear algebra, statistics, and mathematical computations.
- Broadcasting
  This allows for operations on arrays of different shapes efficiently.

- Random Sampling
  This provides a suite of functions for random number generation, which is useful in machine learning and simulations.
- Usage Example
  This is highly optimized for mathematical operations on large datasets, making it an invaluable tool for numerical computing.

4) Sklearn.preprocessing

Purpose: sklearn. preprocessing offers various methods for preparing data for machine learning, including feature scaling and encoding categorical variables.

Key Features:

- LabelEncoder
  This encodes categorical labels into numerical values. It's useful when working with categorical data that needs to be converted to numerical form for algorithms.
- MinMaxScaler
  This scales quantitative characteristics to a certain range (default 0 to 1). It's commonly used to normalize data, improving the performance and convergence speed of many machine learning algorithms.
- Usage Example
  This is essential for preparing data and ensuring that features are appropriately scaled and encoded for effective training in machine learning models.

5) Skfuzzy (skfuzzy as fuzz)

Purpose: Skfuzzy is a fuzzy logic toolkit for Python that enables the creation of fuzzy inference systems.

Key Features:

- Antecedent and Consequent Variables
  This allows the definition of fuzzy input (antecedents) and output (consequents) variables with specified membership functions.
- Fuzzy Rules
  This enables the construction of fuzzy rules (e.g., if-then statements) that describe the relationships between input and output variables.
- Fuzzy Control System Simulation
  This facilitates the creation and simulation of a control system where fuzzy inference is applied to compute outputs based on inputs.
- Usage Example
  This is ideal for implementing fuzzy logic in applications such as control systems, decision-making models, and systems where inputs have uncertainty or imprecision.

4.3.2. Load and Preprocessing of Data

The code in Figure 3 loads training and test datasets from CSV files using *pd.read_*csv. It initializes a LabelEncoder for categorical variables and a MinMaxScaler for numerical values. The categorical columns in the training data are encoded with LabelEncoder. Essential features are extracted by dropping the class column. Test data is encoded in the same way as training data. Numerical features are normalized using MinMaxScaler. After preprocessing, a completion message is printed, and the first few rows of the scaled features are displayed.

4.3.3. Fuzzy Variable Definition

Fuzzy Antecedents, *src_bytes*, and *dst_bytes* are fuzzy input variables representing source and destination byte counts in intrusion detection. They are defined with values ranging from 0 to 5000, with a step of 1. *src_bytes*, and *dst_bytes* are key features in detecting intrusions. Similarly, intrusion_risk is a fuzzy output variable representing the risk level of an intrusion. It is defined with values ranging from 0 to 100, with a step of 1, indicating the range of risk levels from 0% to 100%. These variables are crucial in assessing and responding to potential security breaches.

2. LOAD AND PREPROCESSING OF DATA

[40]:
```
# Load the datasets
train_data = pd.read_csv('FuzzyIDS_dataset/Train_data.csv')
test_data = pd.read_csv('FuzzyIDS_dataset/Test_data.csv')

# Initialize label encoder and scaler
label_encoder = LabelEncoder()
scaler = MinMaxScaler()

# Encode categorical features in the training set
for column in ['protocol_type', 'service', 'flag']:
    train_data[column] = label_encoder.fit_transform(train_data[column])

# Separate features and target in the training data
X_train = train_data.drop(columns=['class'])
y_train = label_encoder.fit_transform(train_data['class'])

# Apply encoding to the test set as well
for column in ['protocol_type', 'service', 'flag']:
    test_data[column] = label_encoder.fit_transform(test_data[column])

# Normalize numerical features in both train and test data
X_train_scaled = pd.DataFrame(scaler.fit_transform(X_train), columns=X_train.columns)
X_test_scaled = pd.DataFrame(scaler.transform(test_data), columns=test_data.columns)

print("Data Preprocessing Complete.")
X_train_scaled.head()
```

Figure 3. Data Loading and Preprocessing for the Intrusion Detection System (IDS)

Data Preprocessing Complete.

[40]:

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_count	dst_host_srv_count	dst_host_same_srv_rate	d
0	0.0	0.5	0.292308	0.9	1.286320e-06	0.000000	0.0	0.0	0.0	0.0	...	0.588235	0.098039	0.17	
1	0.0	1.0	0.630769	0.9	3.824902e-07	0.000000	0.0	0.0	0.0	0.0	...	1.000000	0.003922	0.00	
2	0.0	0.5	0.707692	0.5	0.000000e+00	0.000000	0.0	0.0	0.0	0.0	...	1.000000	0.101961	0.10	
3	0.0	0.5	0.338462	0.9	6.077927e-07	0.001583	0.0	0.0	0.0	0.0	...	0.117647	1.000000	1.00	
4	0.0	0.5	0.338462	0.9	5.213394e-07	0.000082	0.0	0.0	0.0	0.0	...	1.000000	1.000000	1.00	

5 rows × 41 columns

[55]:
```
X_test_scaled.head()
```

[55]:

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_count	dst_host_srv_count	dst_host_same_srv_rate	d
0	0.000000	0.5	0.692308	0.1	0.000000e+00	0.000000	0.0	0.0	0.0	0.0	...	1.000000	0.039216	0.04	
1	0.000000	0.5	0.692308	0.1	0.000000e+00	0.000000	0.0	0.0	0.0	0.0	...	1.000000	0.003922	0.00	
2	0.000047	0.5	0.292308	0.9	3.401281e-05	0.000000	0.0	0.0	0.0	0.0	...	0.525490	0.337255	0.61	
3	0.000000	0.0	0.200000	0.9	5.239592e-08	0.000000	0.0	0.0	0.0	0.0	...	0.011765	0.223529	1.00	
4	0.000023	0.5	0.846154	0.2	0.000000e+00	0.000003	0.0	0.0	0.0	0.0	...	0.113725	0.337255	0.31	

5 rows × 41 columns

Figure 4. Result of Data Preprocessing

4.3.4. Membership Function for Key Input Variables

The membership functions for *src_bytes*, *dst_bytes*, and *intrusion_risk* define categories based on triangular functions. For src_bytes, low values close to 0 belong to the "low" category, while values around 2000 transition out. Medium values around 2500 are fully in the "medium" category, with gradual transitions at 1000 and 4000. High values close to 5000 belong to the "high" category, with transitions starting at 3000. Similarly, *dst_bytes* is categorized as low, medium, or high based on values following the same triangular function.

The intrusion_risk category determines the likelihood of intrusion based on values near 0 for low risk, around 50 for medium risk, and close to 100 for high risk. By using these membership functions, the system can interpret and classify different input values for *src_bytes*, *dst_bytes*, and intrusion_risk effectively.

4.3.5. Fuzzy Rule Definition

This rule provides for reasoning with imprecise values like "high," "medium," or "low," such as in assessing intrusion risk based on byte sizes. The code components involve *ctrl.Rule* function in a fuzzy logic control library, input variables src_bytes and *dst_bytes* representing sent and received bytes categorized as "high," "medium," or "low," and output variable *intrusion_risk* categorized similarly. Rules are defined such that if *src_bytes* and *dst_bytes* are equally "high," intrusion risk is "high," if both are "medium," intrusion risk is "medium," and if both are "low," intrusion risk is "low."

4.3.6. Control System and Simulations for the Three Rules

The code creates a fuzzy control system named *intrusion_ctrl* by combining the three defined rules (rule1, rule2, rule3) using the *ctrl.ControlSystem* function from a fuzzy logic library. This system evaluates intrusion risk based on varying src_bytes and dst_bytes combinations.

A simulation environment named *intrusion_simulation* is then created by using *ctrl.ControlSystemSimulation* for the *intrusion_ctrl* system. This environment allows for running simulations with specific *src_bytes* and *dst_bytes* values, processing them through fuzzy rules, and determining an intrusion risk level. The setup enables testing different scenarios and is crucial for applications like intrusion detection systems, which require continuous risk assessment from uncertain data.

4.3.7. Simulation for Intrusion Detection

The code *intrusion_simulation = ctrl.ControlSystemSimulation(intrusion_ctrl)* creates a simulation environment for the intrusion control system. With this setup, users can input src_bytes and dst_bytes values, run them through fuzzy rules, and get an intrusion risk level as output. This enables testing different scenarios and is crucial for applications like intrusion detection systems that require real-time risk assessment based on uncertain data.

4.3.8. Computed Intrusion Risk Level

The computed intrusion risk level is accessed from the intrusion simulation object after running the intrusion simulation *compute()*. The defuzzified value of intrusion risk is determined based on the src_bytes and dst_bytes input values. The calculated risk level is printed as "*Computed Intrusion Risk Level: 50.00000000000001*". This numeric value signifies the level of risk determined by the fuzzy system. The value of 50.00000000000001 likely corresponds to a "medium" intrusion risk level. This output is the result of assessing input values through fuzzy rules to obtain a clear intrusion risk value, which can be used to make decisions, such as triggering an alert if the risk level exceeds a set threshold.

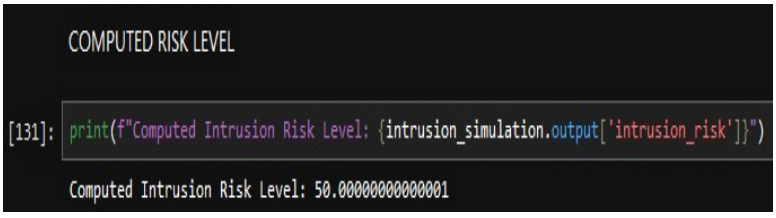

Figure 5. Result of Computed Intrusion Risk Level

International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 81-100, December 2025. **DOI: 10.36079/lamintang.ijai-01202.870**

5. Conclusion

In the two-variable analysis, fuzzy input variables representing the src_bytes and dst_bytes counts were defined with membership functions for low, medium, and high values. The fuzzy rules determined the intrusion risk as high, medium, or low based on combinations of src_bytes and dst_bytes values. The output, intrusion_risk, was computed based on fuzzy rules that assessed risk levels depending on the values of the two input variables. In this experiment, an example scenario yielded a computed risk level of around 50, indicating a medium risk level, which reflects a balanced classification that considers ambiguous input values.

A complex analysis was carried out where additional variables were incorporated: duration, *src_packet_rate*, *dst_packet_rate*, *num_failed_logins*, and *logged_in*. These inputs had tailored membership functions to capture low, medium, and high values, reflecting varying network behaviors. A more comprehensive set of rules was defined to evaluate combinations of these variables and calculate the intrusion risk.

This resulted in more nuanced risk assessments, with an output value of around 49.99, again in the medium-risk range. This summary was drawn based on a series of fuzzy rules that specified how various input values relate to risk, allowing the model to handle multiple dimensions of uncertainty in network traffic data.

The fuzzy logic-based IDS effectively handles uncertain network data by mapping inputs to lexical terms (e.g., "low," "medium," and "high") and using fuzzy rules to compute intrusion risk scores. A two-variable analysis provided simplified risk assessments, while a multi-variable analysis offered improved accuracy by incorporating additional security factors. Both approaches produced medium risk levels in example scenarios, demonstrating the system's conservative and scalable risk categorization. Performance metrics (precision, recall, F-measure) confirm its suitability for real-time traffic analysis. Python integration and machine learning libraries enhance its scalability and adaptability for evolving intrusion patterns.

The research explored the scalability and practical applicability of a fuzzy logic-based IDS for real-time observation. The model demonstrated the ability to manage live data streams and assess risks dynamically, making it suitable for environments where immediate detection and response are crucial. By applying fuzzy logic to intrusion detection, the system can effectively deal with unclear or imprecise information in network traffic. It converts continuous network variables into linguistic terms for a more human-like risk assessment process. The IDS also offers enhanced interpretability and flexibility compared to traditional threshold-based systems, allowing administrators to adjust risk levels based on fuzzy rules. Using the KDD Cup 99 dataset for evaluation further demonstrates the effectiveness of fuzzy-based systems in intrusion detection.

This research lays the groundwork for hybrid IDS models combining fuzzy logic and machine learning to refine accuracy and interpretability. Future research should focus on:

- Adaptive Fuzzy Rules
  Explore dynamic adjustment of fuzzy rules and membership functions using live data or optimization algorithms to enhance flexibility and responsiveness to evolving threats.
- Real-Time Applications
  Test the system with real-time data streams (e.g., Apache Kafka) to enable continuous monitoring and rapid response in high-security environments.
- Explainable AI (XAI)
  Integrate XAI techniques like LIME or SHAP to improve transparency, clarify decisions, and increase trust in the system’s outputs.

These advancements would further refine the model’s adaptability, real-world relevance, and usefulness in cybersecurity.

References

[1] D. Javaheri, S. Gorgin, J. Lee, and M. Masdari, “Fuzzy Logic-Based DDoS Attacks and Network Traffic Anomaly Detection Methods: Classification, Overview, and Future Perspectives”, *Information Sciences,* vol. 626, no. 1, 2023.

[2] F. Jemili, “Intelligent intrusion detection based on fuzzy Big Data classification”, *Cluster Computing,* vol. 26, no. 7, pp. 1-18, 2022.

[3] M. Reji, C. Joseph, K. Thaiyalnayaki, and R. Lathamanju, “Genetic-based Fuzzy IDS for Feature Set

Reduction and Worm Hole Attack Detection", *Computer Systems Science and Engineering,* vol. 45, no. 2, pp. 1265-1278, 2022.

[4] S. Subramani and S. Munuswamy, "Intelligent IDS in wireless sensor networks using deep fuzzy convolutional neural network", *Neural Computing and Applications,* vol. 35, no. 20, pp. 1-20, 2023.

[5] A. Binthiya and S. Ravindran, "Intelligent fuzzy logic-based intrusion detection system for effective detection of black hole attack in WSN", *Peer-to-Peer Networking and Applications*, vol. 17, no. 4, pp. 1-17, 2024.

[6] A. Pinto, L. Herrera, Y. Donoso, and J.A. Gutierrez, "Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure", *Sensors,* vol. 23, no. 5, 2415, 2023.

[7] N. Gupta, V. Jindal, and P. Bedi, "A Survey on Intrusion Detection and Prevention Systems", *SN Computer Science,* vol. 4, no. 5, 2023.

[8] C. Kim, C. So-In, Y. Kongsorot, and P. Aimtongkham, "FLSec-RPL: a fuzzy logic-based intrusion detection scheme for securing RPL-based IoT networks against DIO neighbor suppression attacks", *Cybersecurity*, vol. 7, no. 7, 2024.

[9] V. Hnamte and J. Hussain, "Network Intrusion Detection using Deep Convolution Neural Network", *4th International Conference for Emerging Technology (INCET)*, 2023.

[10] M. Alohali, M. Elsadig, F. Al-Wesabi, M. Al Duhayyim, A.M. Hilal, and A. Motwakel, "Swarm intelligence for IoT attack detection in the fog-enabled cyber-physical system", *Computers & Electrical Engineering*, vol. 108, 108676, 2023.

[11] J. Diaz-Verdejo, J. Munoz-Calle, A. Estapa Alonso, R. Estapa Alonso, and G. Madinabeitia, "On the Detection Capabilities of Signature-Based Intrusion Detection Systems in the Context of Web Attacks", *Applied Sciences,* vol. 12, no. 2, 852, 2022.

[12] A. Belenguer, J.A. Pascual, and J. Navaridas, "GowFed – A novel Federated Network Intrusion Detection System", *Journal of Network and Computer Applications,* vol. 217, 103653, 2023.

[13] A. Kamboj, V.K. Ravindran, and S. Ojha, "A Comparative Analysis of Signature-Based and Anomaly-Based Intrusion Detection Systems", *International Journal of Latest Technology in Engineering, Management & Applied Science,* vol. 14, no. 5, pp. 209-214, 2025.

[14] S. Kushal, B. Shanmugam, J. Sundaram, and S. Thennadil, "Self-healing hybrid intrusion detection system: an ensemble machine learning approach", *Discover Artificial Intelligence,* vol. 4, no. 28, 2024.

[15] H. Asad, S. Adhikari, and I. Gashi, "A perspective-retrospective analysis of diversity in signature-based open-source network intrusion detection systems", *International Journal of Information Security,* vol. 23, pp. 1331-1346, 2024.

[16] M. Shenify, F.A. Mazarbhuiya, and A.S. Wungreiphi, "Detecting IoT Anomalies Using Fuzzy Subspace Clustering Algorithms", *Applied Sciences,* vol. 14, no. 3, 1264, 2024.

[17] A.K. Ramamoorthy and K. Karuppasamy, "Unified Intrusion Detection Framework: Predictive Analysis of Intrusions in Sensor Networks", *Wireless Personal Communications,* vol. 137, no. 3, 2024.

[18] E. Muniyandy, I.G.A. Cruz, M. Farooq, Y. Jaipalreddy, R. Kumar, and V.K. Pandey, "Fuzzy Intrusion Detection Method and Zero-Knowledge Authentication for Internet of Things Networks", *International Journal of Intelligent Systems and Applications in Engineering,* vol. 12, no. 21s, pp. 584-591, 2024.

[19] A. Bengag, A. Bengag, O. Moussaoui, and M. Blej, "A Fuzzy Logic-Based Intrusion Detection System for WBAN Against Jamming Attacks", *Lecture Notes in Electrical Engineering,* vol. 954, 2023.

[20] A. Bengag, A. Bengag, and O. Moussaoui, "Intrusion detection based on fuzzy logic for wireless body area networks: review and proposition", *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 26, no. 2, 1091, 2022.

[21] D.K. Jain, W. Ding, and K. Kotecha, "Training fuzzy deep neural network with honey badger algorithm for intrusion detection in cloud environment", *International Journal of Machine Learning and Cybernetics,* vol. 14, no. 6, pp. 1-17, 2023.

[22] M.G. Raj and S.K. Pani, "Intrusion detection system using combination of deep residual fuzzy network and white shark-dwarf mongoose optimization", *Soft Computing,* 2023

[23] S. Sekar, P.R. Parvathy, G.K. Gupta, T. Rajagopalan, C.C.S. Basavaraddi, K. Padmanaban, and S. Murugan, "Intrusion detection and prevention using Bayesian decision with fuzzy logic system", *International Journal of Electrical and Computer Engineering,* vol. 15, no. 1, 1200, 2025.

[24] A.J. Bamidele, F. Ayo, R. Panigrahi, A. Garg, A.K. Bhoi, and P. Barsocchi, "A Multi-level Random Forest Model-Based Intrusion Detection Using Fuzzy Inference System for Internet of Things Networks", *International Journal of Computational Intelligence Systems,* vol. 16, no. 1, 31, 2023.

[25] Z. Ling, G. Qi, and H. Min, "Intrusion detection using rough-fuzzy set and parallel quantum genetic algorithm", *Journal of High Speed Networks,* vol. 30, no. 1, 2023.

[26] T.M. Ghazal, S.Y. Siddiqui, M.U. Ullah, A. Ali, H.M. Usama, and A. Younas, "Cloud Security Issues Detection Using Fuzzy Logic", *The 8th International Conference on Next Generation Computing,* 2022.

[27] F.S. Alrayes, N. Alshuqayran, M.K. Nour, M.A. Duhayyim, A. Mohamed, A.A.A. Mohammed, G.P. Mohammed, and I. Yaseen, "Optimal Fuzzy Logic Enabled Intrusion Detection for Secure IoT-Cloud Environment", *Computers, Materials & Continua,* vol. 74, no. 3, pp. 6737-6753, 2023.

[28] P. Khordadpour and S. Ahmadi, "FIDS: Fuzzy Intrusion Detection System for simultaneous detection of DoS/DDoS attacks in Cloud computing", *Artificial Intelligence,* 2023.

[29] V. Hnamte and J. Hussain, "Network Intrusion Detection using Deep Convolution Neural Network", *4th International Conference for Emerging Technology (INCET)*, 2023.

[30] M. Almseidin, J. Al-Sawwa, and M. Alkasassbeh, "Anomaly-based Intrusion Detection System Using Fuzzy Logic", *International Conference on Information Technology (ICIT)*, 2021.