International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05 (May) 2026
Received: 10-3-2026 Accepted: 21-4-2026 Published: 07-5-2026

BIG DATA–DRIVEN DISTRIBUTED INTRUSION DETECTION SYSTEM USING SPARK STREAMING

[1] V.KRISHNA REDDY,[2] SK.MOHAMMAD RAFI,[3] BIYYALA NAGENDRA,[4]GURRRAM PAVAN KALYAN,[5]KOVELAKUNTLA SIVA BHARATH,[6]NAGAM RAVI

[1] PROFESSOR & PRINCIPAL, DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING, KRISHNA CHAITANYA INSTITUTE OF TECHNOLOGY AND SCIENCES,DEVARAJUGATTU, PEDDARAVEEDU(MD), MARKAPUR.

[2]ASST., PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, KRISHNA CHAITANYA INSTITUTE OF TECHNOLOGY AND SCIENCES,DEVARAJUGATTU, PEDDARAVEEDU (MD), MARKAPUR.

[3,4,5,6]STUDENT, DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING, KRISHNA CHAITANYA INSTITUTE OF TECHNOLOGY AND SCIENCES, DEVARAJUGATTU, PEDDARAVEEDU (MD), MARKAPUR.

ABSTRACT

A Distributed Intrusion Detection System (DIDS) using Apache Kafka and Spark Streaming presents a scalable and real-time solution for detecting cyber threats in modern network environments. With the rapid growth of distributed systems, cloud computing, and high-speed data generation, traditional intrusion detection systems often struggle to process large volumes of network traffic efficiently and respond to attacks in real time. The proposed system leverages Apache Kafka as a distributed messaging system to collect, buffer, and stream network data from multiple sources, while Spark Streaming is used for real-time data processing and analysis. Machine learning and anomaly detection techniques are integrated into the streaming pipeline to identify suspicious patterns, unauthorized access, and potential cyberattacks. The system supports high throughput, fault tolerance, and low latency, making it suitable for large-scale network infrastructures. Additionally, the distributed architecture ensures scalability and reliability, enabling continuous monitoring and faster response to security threats. Overall, the proposed system enhances cybersecurity by providing efficient, real-time intrusion detection and improved network protection.

KEYWORDS:

Distributed Intrusion Detection System, Apache Kafka, Spark Streaming, Real-Time Processing, Cybersecurity, Machine Learning, Anomaly Detection, Big Data Analytics, Network Security, Stream Processing

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05 (May) 2026
Received: 10-3-2026 Accepted: 21-4-2026 Published: 07-5-2026

I. INTRODUCTION

With the rapid expansion of distributed systems, cloud computing, and high-speed internet technologies, modern networks generate massive volumes of data continuously. This growth has increased the complexity of managing and securing network infrastructures, making them more vulnerable to cyber threats such as unauthorized access, denial-of-service attacks, malware, and data breaches. Traditional Intrusion Detection Systems (IDS), which rely on centralized architectures and static rule-based mechanisms, often struggle to handle large-scale, real-time data streams and fail to detect sophisticated or evolving attack patterns effectively.

To address these challenges, distributed and real-time processing frameworks have become essential for modern cybersecurity solutions. Technologies like Apache Kafka and Apache Spark Streaming enable efficient handling of high-throughput data streams by providing scalable, fault-tolerant, and low-latency processing capabilities. Kafka acts as a distributed messaging system that collects and streams network data from multiple sources, while Spark Streaming processes this data in real time to extract meaningful insights and detect anomalies.

The proposed **Distributed Intrusion Detection System (DIDS)** leverages these technologies to build a scalable and efficient framework for monitoring and analyzing network traffic. By integrating machine learning and anomaly detection techniques into the streaming pipeline, the system can identify suspicious activities and potential intrusions as they occur. This approach allows for faster detection, improved accuracy, and the ability to adapt to new and unknown threats.

II. LITERATURE REVIEW

Recent research in intrusion detection systems (IDS) has focused on improving detection accuracy and scalability by leveraging distributed computing and real-time data processing techniques. Early IDS approaches were primarily signature-based and anomaly-based systems, which relied on predefined rules or statistical thresholds to identify malicious activities. While effective for known attacks, these systems struggled to detect zero-day attacks and generated high false-positive rates in dynamic network environments [1][2].

With the advancement of machine learning, researchers introduced classification algorithms such as decision trees, support vector machines (SVM), and naïve Bayes for intrusion detection. These models improved detection accuracy by learning patterns from historical network traffic data; however, they required extensive feature engineering and

2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05 (May) 2026
Received: 10-3-2026 Accepted: 21-4-2026 Published: 07-5-2026

were limited in handling large-scale, high-velocity data streams [3].

The emergence of big data technologies led to the adoption of distributed frameworks for IDS. Platforms like Apache Hadoop enabled batch processing of large datasets, improving scalability but lacking real-time processing capabilities. This limitation encouraged the development of streaming-based solutions for timely threat detection [4].

Recent studies have explored the use of Apache Kafka and Apache Spark Streaming for building real-time intrusion detection systems. Kafka is used for efficient data ingestion and streaming, while Spark Streaming processes data in near real-time to detect anomalies and attack patterns. These systems demonstrate high throughput, fault tolerance, and scalability, making them suitable for modern network environments [5].

Deep learning techniques such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks have also been applied to intrusion detection. These models can automatically extract features from raw network data and capture temporal dependencies, leading to improved detection performance compared to traditional methods [6].

Additionally, hybrid intrusion detection systems that combine signature-based and anomaly-based approaches have been proposed to enhance detection accuracy. These systems leverage both known attack patterns and behavioral analysis to identify a wider range of threats [7].

Recent research also emphasizes real-time and distributed IDS architectures capable of handling high-speed network traffic and large-scale data. These systems utilize cloud computing and stream processing frameworks to provide scalable and efficient solutions for cybersecurity challenges [8].

Despite these advancements, challenges such as data imbalance, high false-positive rates, and the need for real-time adaptability remain. These issues highlight the need for more robust, intelligent, and scalable distributed intrusion detection systems using advanced streaming technologies and machine learning techniques [9].

III. EXISTING SYSTEM

The existing intrusion detection systems (IDS) are primarily based on **centralized architectures** and traditional detection techniques such as signature-based and anomaly-based methods. Signature-based systems detect attacks by comparing network traffic against a database of known attack patterns. While they are effective in identifying previously known threats, they fail to detect new or unknown attacks (zero-day

2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05 (May) 2026
Received: 10-3-2026 Accepted: 21-4-2026 Published: 07-5-2026

attacks) and require frequent updates of signature databases.

Anomaly-based systems, on the other hand, identify deviations from normal network behavior using statistical methods or basic machine learning models. Although these systems can detect unknown attacks, they often generate a high number of false positives, making it difficult for security analysts to distinguish between normal and malicious activities.

Many existing systems rely on batch processing frameworks such as Apache Hadoop, which are designed to handle large volumes of data. However, these systems are not suitable for real-time intrusion detection as they process data in batches, leading to delays in identifying and responding to security threats.

Traditional IDS implementations also face limitations in handling **high-speed and large-scale network traffic**. As modern networks generate massive amounts of data continuously, centralized systems become bottlenecks, resulting in reduced performance and scalability issues.

Furthermore, existing systems often lack **real-time data streaming and processing capabilities**, making them less effective in detecting fast-evolving cyberattacks such as distributed denial-of-service (DDoS) attacks and advanced persistent threats (APTs). They also struggle to integrate data from multiple distributed sources, limiting their ability to provide a comprehensive view of network activities.

Another major limitation is the insufficient use of advanced analytics and machine learning techniques, which restricts the system's ability to adapt to new attack patterns and evolving threats. Additionally, these systems may not provide efficient fault tolerance or reliability in distributed environments.

Overall, while existing intrusion detection systems provide basic security functionalities, their limitations in scalability, real-time processing, adaptability, and accuracy highlight the need for a distributed and streaming-based approach using technologies like Apache Kafka and Apache Spark Streaming.

IV. PROPOSED SYSTEM

The proposed system is a **Distributed Intrusion Detection System (DIDS)** that leverages modern stream-processing technologies to provide real-time, scalable, and efficient detection of cyber threats in large-scale network environments. The system is designed to overcome the limitations of traditional centralized IDS by adopting a distributed architecture that integrates data ingestion, real-time processing, and intelligent threat detection.

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05 (May) 2026
Received: 10-3-2026 Accepted: 21-4-2026 Published: 07-5-2026

In this system, network traffic data is collected from multiple distributed sources such as servers, routers, firewalls, and endpoints. The collected data is streamed using Apache Kafka, which acts as a high-throughput, fault-tolerant messaging system. Kafka efficiently handles continuous data streams, ensuring reliable data delivery and buffering between producers (data sources) and consumers (processing units).

The streamed data is then processed in real time using Apache Spark Streaming. Spark Streaming performs data transformation, filtering, and analysis on incoming data streams, enabling the system to detect suspicious activities with minimal latency. The processing pipeline supports both batch and micro-batch processing, ensuring flexibility and efficiency.

The core detection engine incorporates **machine learning and anomaly detection techniques** to identify malicious patterns in network traffic. Supervised learning models are used to detect known attack signatures, while unsupervised methods such as clustering and anomaly detection identify unknown or abnormal behaviors. This hybrid detection approach improves accuracy and reduces false positives.

Additionally, the system includes a **distributed storage layer** (such as HDFS or NoSQL databases) to store processed data, logs, and detected alerts for further analysis and auditing. A visualization dashboard is provided for security analysts to monitor network activities, view alerts, and analyze threat patterns in real time.

The proposed system ensures **scalability** by distributing workloads across multiple nodes, **fault tolerance** through data replication and recovery mechanisms, and **low latency** for immediate threat detection. It also supports continuous learning by updating models with new data to adapt to evolving cyber threats.

Overall, the proposed Distributed Intrusion Detection System provides a robust, intelligent, and scalable solution for modern cybersecurity challenges, enabling efficient real-time monitoring and proactive threat detection in distributed network environments.

V. METHODOLOGY

The methodology of the proposed **Distributed Intrusion Detection System (DIDS) using Apache Kafka and Spark Streaming** follows a structured pipeline that enables real-time data collection, processing, and intelligent threat detection in a distributed environment.

Initially, network data is collected from multiple distributed sources such as routers, servers, firewalls, and endpoint devices. This data includes packet-level information, logs,

2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05 (May) 2026
Received: 10-3-2026 Accepted: 21-4-2026 Published: 07-5-2026

connection details, and user activities. The collected data is continuously streamed into Apache Kafka, which acts as a messaging layer to handle high-throughput data ingestion. Kafka ensures reliable, fault-tolerant, and scalable data streaming by organizing data into topics and partitions.

Once the data is ingested, it is processed in real time using Apache Spark Streaming. Spark Streaming consumes data from Kafka and performs preprocessing tasks such as data cleaning, filtering, normalization, and feature extraction. Important features such as packet size, protocol type, connection duration, and traffic patterns are derived to represent network behavior effectively.

After preprocessing, machine learning and anomaly detection models are applied to analyze the streaming data. Supervised learning algorithms such as decision trees, random forests, and support vector machines (SVM) are used to detect known attack patterns based on labeled datasets. In parallel, unsupervised methods such as clustering and anomaly detection techniques (e.g., Isolation Forest, Autoencoders) are employed to identify unknown or suspicious behaviors in real time.

The system uses a hybrid detection approach where results from multiple models are combined to generate a final classification or risk score for each network event. If the risk score exceeds a predefined threshold, the event is flagged as a potential intrusion. Alerts are generated and forwarded to the monitoring system for further investigation.

Detected events and processed data are stored in a distributed storage system such as HDFS or NoSQL databases for logging, auditing, and future analysis. A visualization dashboard is provided to display real-time alerts, system performance, and network activity trends, enabling security analysts to make informed decisions بسرعة.

Finally, the system is evaluated using performance metrics such as accuracy, precision, recall, F1-score, and detection rate. A continuous learning mechanism is incorporated to update the models with new data, ensuring adaptability to evolving cyber threats. This methodology ensures a scalable, efficient, and real-time intrusion detection system capable of handling modern network security challenges.

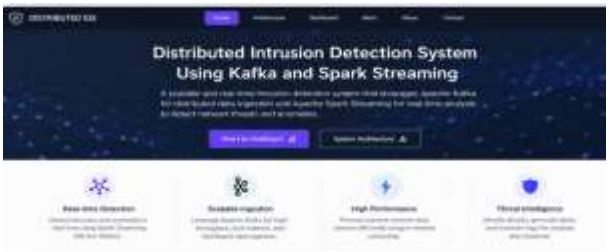
VI. SYSTEM MODEL

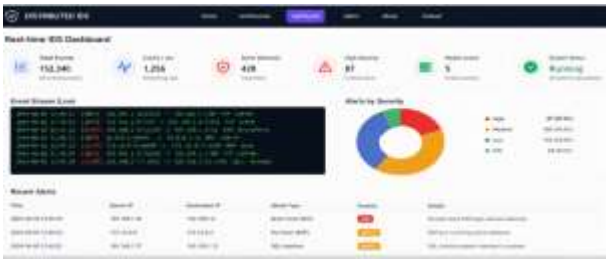
System Architecture

2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 05 (May) 2026
Received: 10-3-2026 Accepted: 21-4-2026 Published: 07-5-2026

VII. RESULTS AND DISCUSSIONS





VIII. CONCLUSION

The proposed **Distributed Intrusion Detection System (DIDS) using Apache Kafka and Apache Spark Streaming** provides a scalable and efficient solution for detecting cyber threats in modern network environments. By leveraging distributed streaming technologies, the system is capable of processing large volumes of network data in real time, ensuring timely detection of potential intrusions.

The integration of machine learning and anomaly detection techniques enhances the system's ability to identify both known and unknown attack patterns with improved accuracy. Unlike traditional IDS, the proposed approach reduces false positives and adapts to evolving cyber threats through continuous learning mechanisms.

Furthermore, the distributed architecture ensures high throughput, fault tolerance, and low latency, making the system suitable for large-scale and high-speed network infrastructures. The use of real-time analytics enables proactive monitoring and faster response to security incidents, thereby strengthening overall network security.

In conclusion, the proposed system represents a significant advancement in intrusion detection by combining big data technologies with intelligent analytics. It offers a robust, flexible, and scalable framework that can effectively address the challenges of modern cybersecurity and protect distributed systems from emerging threats.

IX. FUTURE WORK:

The proposed **Distributed Intrusion Detection System (DIDS) using Apache Kafka and Spark Streaming** can be further enhanced to improve its intelligence, scalability, and real-world applicability. Future work can focus on integrating advanced deep

learning techniques such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based models to improve detection accuracy, especially for complex and evolving cyber threats. These models can better capture temporal and spatial patterns in network traffic.

Another important direction is the adoption of **Graph Neural Networks (GNNs)** to analyze relationships between network entities, enabling the detection of sophisticated attacks such as coordinated intrusions and lateral movements within networks. This would enhance the system's ability to identify hidden attack patterns that traditional methods may miss.

The system can also be extended by incorporating **real-time adaptive learning mechanisms**, where models continuously update themselves using streaming data. This will allow the system to quickly adapt to new attack strategies and reduce dependency on static training datasets.

Future improvements may include the integration of **edge computing**, where intrusion detection components are deployed closer to data sources (e.g., IoT devices or edge nodes). This will reduce latency, improve response time, and enhance security in distributed and resource-constrained environments.

Additionally, implementing **explainable AI (XAI)** techniques can provide transparency in model predictions, helping security analysts understand why a particular activity is flagged as malicious. This is crucial for building trust and ensuring compliance with security standards.

Scalability can be further improved by deploying the system on cloud-native platforms using containerization technologies such as Docker and orchestration tools like Kubernetes. This will enable efficient resource management and seamless scaling based on network load.

XI. REFERENCES

[1] J.V.ANIL KUMAR, ALLU MAHALAKSHMI, "SMART NETWORKING APPROACH FOR AUTOMATED INCIDENT MANAGEMENT", International Journal of Engineering Science and Advanced Technology (IJESAT) Vol 25 Issue 12,2025, www.ijesat.com, https://doi.org/10.64771/ijesat.2025.047, Page 384 to 392, ISSN:2250-3676, 2025.

[2] Jajam Venkata Anil Kumar, Dr. G. Charles Babu, "Automating Content Utilizing Big Data Innovations", *Journal of Advances and Scholarly Researches in Allied EducationVol. 15, Issue No. 9, October-2018, ISSN 2230-7540,* IIFS : 1.6

(2014), INDEX COPERNICUS : 49060 (2018), IJINDEX : 3.46 (2018), pp.635-639, 2018.

[3] Jajam Venkata Anil Kumar, Dr. G. Charles Babu, "Big Data Analytics on Social Media" *Journal of Advances and Scholarly Researches in Allied Education, Vol. XII, Issue No. 23, October-2016, ISSN 2230-7540,* IIFS : 1.6 (2014), INDEX COPERNICUS : 49060 (2018), IJINDEX : 3.46 (2018), pp. 389-393,2016.

[4] White, T., "Hadoop: The Definitive Guide," O'Reilly Media, 2012.

[5] Kreps, J., Narkhede, N., and Rao, J., "Kafka: A Distributed Messaging System for Log Processing," *NetDB*, 2011.

[6] Zaharia, M., Das, T., Li, H., et al., "Discretized Streams: Fault-Tolerant Streaming Computation at Scale," *SOSP*, 2013.

[7] Hochreiter, S. and Schmidhuber, J., "Long Short-Term Memory," *Neural Computation*, 1997.

[8] Goodfellow, I., Bengio, Y., and Courville, A., "Deep Learning," MIT Press, 2016.

[9] Chandola, V., Banerjee, A., and Kumar, V., "Anomaly Detection: A Survey," *ACM Computing Surveys*, 2009.