SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I7P134

Volume 12 Issue 7, 428-442, July 2025
© 2025 Seventh Sense Research Group®

Original Article

BSSA-BiT: A Big Data-Based IoT Intrusion Detection Model

S. Ravishankar[1], P. Kanmani[2]

[1]*Department of Computer Science, Sona College of Arts and Science, Salem, Tamil Nadu. India.*
[2]*Department of Computer Science, Thiruvalluvar Government Arts College, Rasipuram, Tamil Nadu. India.*

[1]*Corresponding Author : ravirohan83@gmail.com*

Received: 18 May 2025 Revised: 19 June 2025 Accepted: 19 July 2025 Published: 31 July 2025

Abstract - *The size of the Internet and network traffic is increasing constantly, with data generated at an extremely fast rate in petabytes. This data can be classified as Big Data (BD) due to its substantial volume, veracity, Velocity, and variety. The proliferation of usage is accompanied by an increase in security risks to networks like the Internet of Things (IoT). Identifying intrusions in a BD context is challenging. Numerous Intrusion-Detection Systems (IDSs) have been developed for various network attacks; however, most of these IDSs are either incapable of identifying unknown attacks or cannot respond. Deep Learning (DL) algorithms, recently utilized for extensive BD analysis, have demonstrated exceptional performance and efficiency in detecting intrusions. Hence, this research proposes a BD-based IoT intrusion detection model using the DL algorithm with Apache Spark for attack detection and classification. The developed research model incorporates the Binary Salp Swarm Algorithm (BSSA) technique for feature selection and the Bidirectional Transformer (BiT) method for attack detection and classification. For training and evaluation, the CIC-IoT-23 BD dataset is collected and used. Using Apache Spark, the data is preprocessed with multiple preprocessing phases such as data cleaning, normalization, oversampling, and encoding. The BSSA technique selects the most optimal features that help the BiT classifier for accurate attack detection, minimizing dimensionality and enhancing learning efficiency. The BSSA-BiT model attained 98.85% accuracy, 98.59% detection rate, 98.94% precision, and 98.70% F1-score in multiclass classification, and compared to other models, it outperformed and demonstrated as an effective IDS model.*

Keywords – *Big Data, Intrusion detection, IoT, Deep Learning, BSSA, BiT, Apache spark.*

1. Introduction

In the current digital world, big data has become a transformative technology, altering the methods by which organizations/businesses gather, store, and analyze large datasets. The substantial rise in data generated from diverse sources, including social media, sensors, and autonomous vehicles, has transformed the BD analytics necessary for businesses and organizations worldwide. The worldwide BD industry is anticipated to experience significant growth, with revenue predictions reaching 473.6 billion USD by 2030, indicating an annualized increase of 12.7% from 2020 to 2030. Current projections predict a substantial surge in the generation of data, where the global output is anticipated to reach 175 zettabytes by 2025. This rapid increase underscores the rising significance of BD as a key tool for analyzing, managing, and extracting insights from the vast quantity of data [1]. BD plays a vital role in cybersecurity. Particularly in areas like anomaly detection, intrusion detection, spoofing and spam detection, ransomware and malware detection, cloud security, and code security. Cybersecurity analytics in BD is progressively emerging as a critical domain of study and implementation focused on securing networks, data, and computers from unauthorized access by evaluating security event data through BD techniques and technology. The cybersecurity analytics in BD are versatile and suitable for various types of attacks, such as IDSs or alert correlators; conversely, other systems are designed for specific threats, such as malware and phishing detection. Cybersecurity analytic systems like IDS are deployed with and without BD technologies [2].

In the modern technical research field, BD is crucial, emphasizing the analysis, extraction, and processing of significant data from large and complex datasets. The core concept of BD analytics is closely associated with modern technologies, particularly the IoT, Artificial Intelligence (AI), Machine Learning (ML), and DL. The major domains of BD applications encompass Cybersecurity, Healthcare, Education, Market Analysis, Supply Chain and Transportation, Smart Cities, Earth Sciences, Media, Industry, etc. Figure 1 depicts the stages involved in the processing of BD.

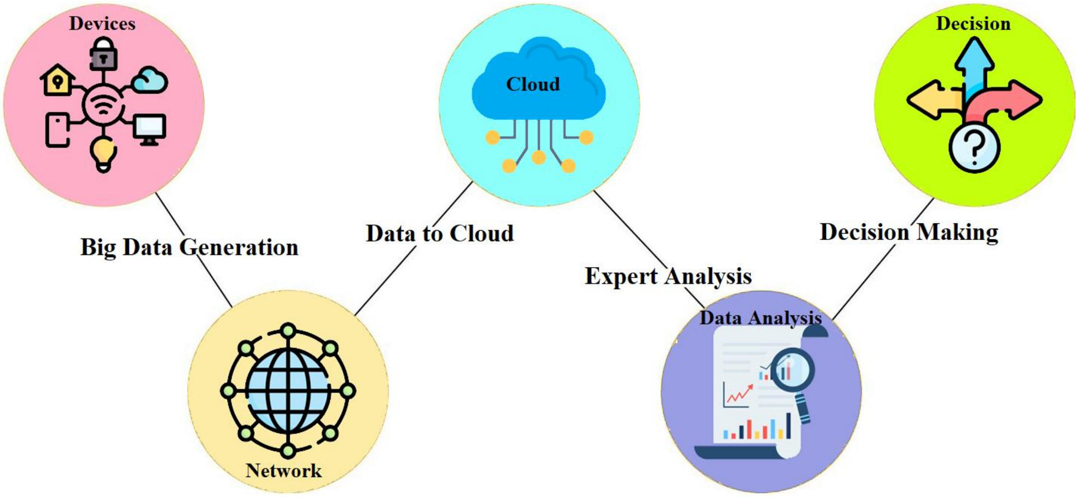

Fig. 1 Structure of big data analysis

The analysis of network traffic, system activities, and records for intrusion detection is a critical subject in cybersecurity. The analytical methods of IDS encompass misuse detection (signature detection), anomaly detection, and specification-based detection (integration of misuse and anomaly detection). Anomaly detection controls standard behaviour patterns of users and detects intrusions by recognizing deviations from typical activities. Nonetheless, it could result in a significant false-alarm rate, since new user activities not encompassed within the user behaviour patterns could be misclassified as attacks. IDS systems based on anomaly detection are capable of monitoring network traffic; nevertheless, they frequently face the significant risks of false positives. The misuse-based IDS technique finds intrusions by comparing security activity with previously recorded attacks. This intrusion detection system is effective solely against established patterns of behaviour and is incapable of detecting new or unfamiliar attacks. The database requires continual updating to incorporate new attack patterns. IDS based on misuse detection can identify spyware, malware, and various malicious activities [3].

Network traffic, system activities, and logs are typically classified as BD. The analytics of BD and associated technologies enhance cybersecurity by constantly monitoring the flow of data, analyzing incidents, discovering anomalies and variations in network traffic, and detecting attackers. As depicted in Figure 2, the features of BD can be delineated by the "7 Vs": Volume, Velocity, Variety, Variability, Valence, Veracity, and Value [4]. Volume signifies the magnitude of data, Velocity pertains to the speed of data generation and acquisition, and Variability indicates irregularities and unpredictable variations in data throughout time [5]. Variety refers to the diversity of data formats, types, representations, and semantic interpretations. Valence denotes the degree of connectivity; two data elements are interconnected when they are associated. Veracity pertains to the precision, authenticity, and dependability of data. Defective datasets exhibit a lack of veracity in the context of BD and are of inferior quality. Due to their low veracity, they consequently possess small Value [6].

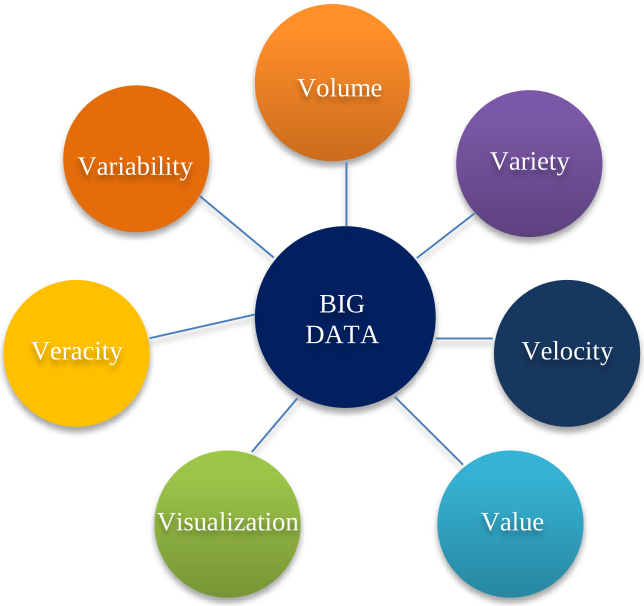

Fig. 2 Seven Vs Big data

IoT networks encompass the connectivity of actuators, sensors, and services, necessitating substantial hardware and software for the execution of BD analytics and applications related to cybersecurity [8]. Cyber-attacks provide escalating difficulties in accurately identifying intrusions, compromising data integrity, confidentiality, and availability [8].

1.1. Problem Statement

The design of IDSs is significantly complicated by the evolution of malicious software, referred to as malware. The primary challenge in identifying unknown and obscured malware is that intruders employ many methodologies to avoid detection by IDS. Thus, the technical ability of harmful attacks has escalated. Conventional cybersecurity systems fail to manage the increasing abundance of data that users, devices, and networks generate. Networking devices and applications produce substantial volumes of diverse data. Organizations consistently carry out communication and data exchange with others. Data is sourced from a variety of origins, including open-access government databases, publicly accessible social media communications, beta testers seeking vulnerabilities, and security experts doing assurance audits on widely utilized software. Organizations must process, analyze, and correlate all this information in real time to protect against cyber threats. BD technologies enable the analytical capabilities required to link unrelated occurrences. To address the rise in cyberattacks and enhance a network's security status, organizations are now using DL and ML algorithms for attack and threat detection [9]. DL is a domain of ML that has contributed to the growing prominence of AI-based methodologies. DL can process unprocessed data without the need for feature extraction. It can identify unique patterns among the incoming data, accommodate new file formats, and detect unknown attacks. Numerous researchers have utilized DL methodologies for security-related tasks [10]. Thus, a DL-based IDS is proposed in this research work to secure BD in IoT systems.

1.2. Research Objectives

The novelty of the developed research model is based on the integration of two different algorithms. The first is BSSA for optimal feature selection, and the next is the BiT model for accurate attack classification. Both these algorithms are integrated and implemented on the Apache Spark BD framework. This novel integration of the model provides effective handling of BD network traffic while improving the detection and classification accuracy on both binary and multiclass tasks. The major contributions of this proposed research work are defined in the following:

- A hybrid BSSA-BiT IoT intrusion detection model is developed for robust attack detection and classification.
- The BSSA is employed for optimal feature selection, and the BiT is employed for attack classification.
- This BSSA-BiT model is implemented on the Apache Spark BD platform to process a huge volume of network traffic data.
- The BSSA-BiT model is assessed on both the binary and multiclass classification based on multiple performance indicators.
- The evaluated results are validated by comparing the BSSA-BiT model's results with the current IDS models discussed in this research.

The paper is organized into the subsequent sections. Section II succinctly examines the current models relevant to the research study. Section III encompasses the implementation of the developed research methodology. Section IV highlights the experimentation findings of the research methodology and a comparison with current models. The final section concludes the research with an overview of the findings and recommendations for subsequent research initiatives.

2. Related Works

This section presents a review of current works applied to improving big data-based intrusion detection. All the reviewed current models are critically analyzed and presented in Table 1. The critical analysis includes their advantages and limitations. The review of the related works is as follows: An Elephant Herd Optimizer-based Finite Dirichlet Mixture Method (EHO-FDMM) was proposed in [11] for BD intrusion detection. This framework comprised three modules: capture and logging, pre-processing, and an IDS approach utilizing the EHO-FDMM. The UNSW-NB15 and NSL-KDD datasets were utilized to evaluate the performance of this system. The findings indicated that statistical analysis facilitated the selection of the optimal model that accurately matches the network data. The EHO-FDMM-based IDS provided a reduced False Alarm Rate (FPR) and an enhanced Detection Rate (DR).

A full-stream BD system with optimized DL for cybersecurity (FSBDL) was proposed in [12]. The model utilized two concurrent optimizers, such as Adam and RMSprop. This Hyper-Parallel optimization (HPO) approach was developed to improve efficiency and stability. The optimized Convolutional Neural Network (CNN) under the model attained elevated accuracy in real-time attack detection while maintaining reliability. A deep CNN–Weighted Deep Long Short-Term Memory (WDLSTM) IDS model was developed in [13] for a big data model. The deep CNN recovered significant attributes from the attack data, using its advantage due to sharing weights. A dropout strategy was employed to disregard certain neurons during model training to mitigate overfitting randomly. Subsequently, the developed network model was employed to discern dependencies within the extracted attributes and address the issue of data imbalance. At last, the model's hyperparameters were refined through iterative experimentation. Experimental findings demonstrated better results, with 97.1% accuracy.

A Network IDS (NIDS) technique based on deep neural networks within a BD environment was developed in [14]. Feature reduction and anomaly probability output were fundamental to both levels. Subsequently, a CNN, comprising a layer called down-sampling and a feature extraction layer consisting of a convolutional layer, achieved input correlation by the incorporation of bidirectional LSTM. Next, a pooling layer was incorporated following the convolution layer to extract essential features based on various sampling criteria. This enhanced the overall efficacy of the NIDS model. A hybrid Stacked Autoencoder (SAE)–

Support Vector Machines (SVM) framework for a rapid and effective cybersecurity IDS was proposed in [15]. The model employed a network for latent feature extraction. The study utilized various classification-based IDS methods like SVM, Random Forest (RF), Decision Tree (DT), and Naive Bayes (NB) for performing IDS in extensive network data. The efficacy of all the models was evaluated utilizing the BD analytics tool Apache Spark. The model's results highlighted that the accuracy was increased and the execution time was reduced.

The research in [16] developed an IDS utilizing the BD platform, Apache Spark. Apache Spark was utilized in conjunction with its ML library (MLlib) and the BoT-IoT dataset. The IDS was subsequently assessed and analyzed using F-Measure, which was the conventional method for evaluating unbalanced data. Two rounds of testing were conducted: one utilizing a partial dataset to mitigate bias, and the other employing the complete BoT-IoT dataset to investigate BD and ML abilities within a security context.

The RF algorithm has shown superior performance in classification for the dataset. A data flow utilizing structured streaming using Apache Spark, Apache Kafka, and MongoDB was developed in [17] capable of real-time adaptation to evolving attack patterns and classification of attacks within extensive IoT networks. Upon detecting concept drift, the model retrained itself using the data samples that triggered the drifts alongside the representative subsamples with the model's prior training data. The classifier was assessed using the most recent dataset, IoT23. The training duration of the distributed RF algorithm was assessed by altering the count of cores in the Apache Spark platform, yielding improved outcomes.

A security model based on distributed computing to protect BD systems was developed in [18]. The Ensembled Multi-Binary Attack Model (EMBAM) provided a distinctive anomaly-based approach to identify both normal behaviour and anomalous attacks, such as network threats. EMBAM integrated numerous binary classifiers into a unified model by stacking. The fundamental binary model was a DT classifier with hyperparameters refined by the grid search technique. The EMBAM classifier was evaluated, demonstrating better performance and a high detection rate.

A streaming Sliding Window Local Outlier Factor Coreset Clustering Method (SSWLOFCC) was proposed in [19]. The method employed a clustering-based technique for anomaly identification utilizing BD technology, including Flume, BroIDS, Spark Streaming, Kafka, Spark MLlib, HBase, and Matplotlib. It was assessed to validate its effectiveness. The evaluation results have significantly demonstrated the effectiveness of the method with a markedly increased accuracy rate.

The research in [20] developed a Java system to establish a model with a substantial flow of data that identified attacks in a distributed framework. The model utilized an operating system with distributed components for data acquisition, analysis, and storage. The findings demonstrated that external DDoS attacks were promptly identified. The failure of the single-point problem was resolved, mitigating the limitation in data processing capacity. ML methodologies, including K-means clustering, DT, Bayesian classifiers, SVM, and ensemble learning techniques such as Bagging, Boosting, and Stacking, were used in [21]. DL methodologies utilized CNN for feature extraction, Recurrent Neural Networks (RNN) for temporal data analysis, and approaches such as deep transfer learning and multitask learning to improve detection precision. These integrated methodologies seek to enhance the precision, efficacy, and resilience of network traffic detection of anomalies and security measures based on BD.

An IDS model utilizing BD mining, incorporating fuzzy rough set theory, Generative Adversarial Networks (GAN), and CNN was proposed in [22]. A fuzzy rough set-based technique was introduced for feature selection in BD through IoT applications. Subsequently, CNN's effective feature extraction abilities were leveraged to develop an IDS based on the chosen features. Additionally, an approach was applied for intrusion detection across diverse circumstances by integrating CNN and GAN. Simulation findings indicated that the method's performance was superior. An ensembled framework was developed in [23] to assess dimensionality reduction in the IDS model, with several combinations evaluated and processed on datasets. BD methodologies have been employed to process and analyze extensive datasets, facilitating a comprehensive data analysis procedure. The ensemble methodology used the bagging technique in conjunction with boosting techniques. Additionally, a DT has been incorporated into this methodology. The experimentation demonstrated that the evaluated ensemble models exhibited superior performance.

The research in [24] utilized Device Type Identification and Device-based IDS (DIDS) techniques. The DIDS learning model integrated the classification of unidentified attacks to manage computational demands in extensive networks. The model also enhanced throughput while maintaining a low false alarm rate. The performance revealed that RF was the most effective DIDS algorithm and excelled in recognizing device types. The utilization of memory data for malware detection was utilized in [25]. Malware detection was executed utilizing diverse DL and ML methodologies within a BD framework featuring memory data. The research was conducted using PySpark on the Apache Spark BD platform. Binary classification was conducted with RF, DT, Gradient Boosted Tree, NB, Logistic Regression (LR), Multilayer Perceptron, SVM, Deep Feedforward Neural Network, and LSTM techniques.

The LR technique yielded the highest efficacy in malware identification. The research in [26] utilized event profiles and Artificial Neural Networks (ANN) to present the DL-based Security Information Systems (DL-SIS). The research enhanced the detection of anomalies by distilling BDsets into event profiles and employing DL-based detection techniques. The model enabled security analysts to respond promptly and efficiently to urgent security alerts by analyzing data over extended durations. The model achieved the maximum detection accuracy, although it exhibited poor performance in the high-dimensional dataset.

Table 1. Critical analysis of analyzed related works

Ref	Model	Application	Advantages	Disadvantages
[11]	EHO-FDMM	Big Data Intrusion Detection (UNSW-NB15, NSL-KDD)	Reduced false positive rate; Improved detection accuracy.	Optimization model was computationally expensive; it needs fine-tuning.
[12]	FSBDL	Cybersecurity (real-time attack detection)	High accuracy and stability in real-time detection.	Requires large computational resources for dual-optimizer training.
[13]	Deep CNN + Weighted DLSTM + Dropout + Hyperparameter tuning	IDS for big data	97.1% accuracy; Overfitting was reduced using Dropout.	Deep models are resource-intensive; Difficult to interpret.
[14]	CNN + Bidirectional LSTM + Pooling Layer	Network-based IDS in big data	Enhances feature extraction and sequence modelling.	Complexity increases training time; High memory usage.
[15]	SAE + SVM + Apache Spark	Efficient IDS in high-volume traffic	High speed and accuracy with Spark; Multiple classifiers tested.	Requires expertise in Spark configuration and tuning.
[16]	Apache Spark + MLlib + RF on BoT-IoT dataset	Big data intrusion detection in IoT	Balanced testing, Spark enables scalability.	Suffer from class imbalance if not handled properly.
[17]	Apache Kafka + Spark Streaming + MongoDB + RF + Concept Drift Handling	Real-time IoT attack detection	Adaptable to evolving attacks; Distributed processing.	Requires robust stream management; Complex setup.
[18]	EMBAM	Big data anomaly detection	High detection rate; Effective ensemble model.	Ensemble increases computational complexity.
[19]	SSWLOFCC	Real-time anomaly detection in network data	Improved accuracy; Real-time adaptation.	Cluster maintenance over the stream can be costly.
[20]	Java-based distributed system for intrusion detection	Distributed DDoS detection	No single-point failure; Scalable.	Limited ML integration; the General framework lacks algorithmic novelty.
[21]	ML: K-means, SVM, DT, Bagging, Boosting; DL: CNN, RNN, Transfer Learning	Big data anomaly and threat detection	Robust feature analysis; Ensemble enhances resilience.	High dimensionality can impact performance.
[22]	Fuzzy Rough Set + CNN + GAN	IoT big data IDS	Superior performance; Effective feature selection.	GAN training is unstable; Model complexity is high.
[23]	Ensemble (Bagging + Boosting + DT)	Big data-based IDS with dimensionality reduction	Better performance with large datasets.	Multiple models can slow execution.
[24]	Device-based IDS + RF + DL/ML	Device type classification and anomaly detection	High accuracy (95%); Manages large-scale networks.	Do not generalize to unknown device types.
[25]	DL & ML (RF, DT, GBT, LR, SVM, LSTM) on PySpark	Malware detection in big data memory traces	Logistic Regression was most effective, with Real-time capability.	Feature engineering needed; Potential overfitting.
[26]	DL-SIS (Event Profiling + ANN)	Big data anomaly detection via event logs	High detection accuracy; Effective long-term analysis.	Poor performance on high-dimensional data.

2.1. Research Gap Analysis

After analyzing the related works, there are a few research gaps from the recent advancements in IDS models based on BD using diverse ML and DL methodologies. Most of these models aimed to enhance the detection accuracy and system scalability. However, these models neglect adaptability for evolving IoT attacks and variations.

The majority of the models lack a feature selection process and are dependent on computationally intensive models. This led to latency and inefficiencies in detecting attacks. Additionally, few models utilize lightweight and effective optimization techniques appropriate for managing high-dimensional data in IoT networks. These gaps highlight the need for robust, scalable and efficient IDS models for large-scale IoT data using effective optimization and classification algorithms.

3. Materials and Methods

This research proposed a BD-based IoT intrusion detection model using the DL algorithm with Apache Spark for attack detection and classification. The developed research model incorporates the BSSA technique for feature selection and the BiT method for attack detection and classification. The developed research model's workflow was depicted in Figure 3. As shown in the figure, the CIC-IoT-23 BD dataset is initially collected and used for training and testing the research model. Using Apache Spark, the data is preprocessed with multiple preprocessing phases, such as data cleaning, normalization, oversampling, and encoding, to make the data in a standard manner and appropriate for the developed model.

The preprocessed data is divided into training and test subsets, which are used for training the model and evaluation. The binary variant of the SSA model is applied to select features from the training data. This BSSA technique selects the most optimal features that help the classifier for accurate attack detection, minimizing dimensionality and enhancing learning efficiency. The selected features are applied to train the BiT model. This DL architecture is capable of capturing contextual relations in sequential data and is optimal for complex intrusion detection tasks. At last, the trained model was evaluated using common evaluation metrics to assess its performance in detecting different attack types.

3.1. Dataset Details

The CIC-IoT 2023 dataset comprises a range of recent IoT attacks. This collection contains communications from 105 authentic IoT devices and includes 33 distinct attack methods. To enhance classification performance, these attacks were categorized into seven distinct categories: DoS, DDoS, Spoofing, Brute Force, Recon, Mirai, and Web-based. This CIC-IoT-2023 dataset comprises 46 features.

Table 2. Dataset distribution

Attack Classes	No. of Records	Distribution %
Benign/Normal	1098195	2.35
Spoofing	486504	1.04
Reconnaissance	354565	0.76
Bruteforce	13064	0.03
WebBased	24829	0.05
DoS	8090738	17.33
Mirai	2634124	5.64
DDoS	33984560	72.79

As shown in Table 2, the attack category with the most records is DDoS, which overflows networks or devices with an overwhelming traffic volume, resulting in disruptions and rendering services inaccessible. The next dominant class, DoS, resembles DDoS but generally originates from a single source and similarly seeks to impair service availability. Brute Force seeks to obtain unauthorized access by systematically testing many password combinations. Spoofing leads devices by disguising itself as authentic entities, resulting in data exfiltration or virus distribution. Reconnaissance attacks collect network data to find vulnerabilities. Web-based attacks leverage vulnerabilities in web applications to gain unauthorised access unauthorized. The Mirai attack exploits IoT, converting devices into bots for extensive attacks like DDoS. This dataset offers a thorough analysis of the prevailing IoT attacks. The DDoS category contains a large number of instances, followed by DoS and Mirai. Remaining classes, like Recon, Spoofing, Brute Force, and Web-based, exhibit significantly limited samples [27].

3.2. Apache Spark

Apache Spark (APS) is an open-source initiative intended to parallel process BD sets. It is designed using the Scala programming language. The fundamental design is the Resilient Distribution Dataset (RDD). RDDs are dispersed, adaptable, and resilient structures. APS does data processing in memory. This capability enables speed processing, surpassing MapReduce's ability to operate on memory. APS comprises the components of Spark SQL, Spark Core, Spark Streaming, Horovod, and GraphX. The Spark Core serves as the foundational framework for every component. Spark Streaming, Spark SQL, Horovod, and GraphX were the principal libraries of APS. Spark SQL handles structured information, whereas Spark Streaming is used for real-time data processing: Horovod, a DL library for Spark. Network and graph analysis were conducted using the GraphX package. These library tools could be utilized concurrently within an individual project. Spark offers multi-language assistance with project implementation. Applications could be developed on Spark utilizing R, Scala, Python, and Java programming language tools. APS could utilize the Hadoop Distributed File Systems (HDFS) for storage, and can be combined with various BD technologies.

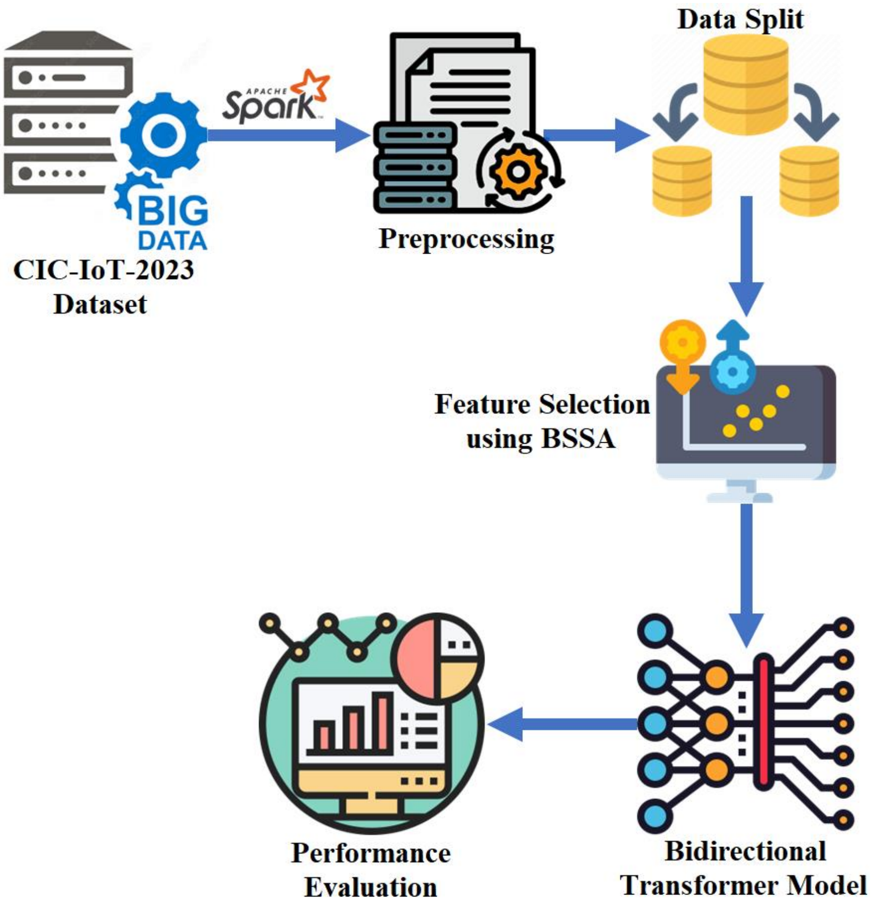

Fig. 3 Proposed Big Data-based IoT intrusion detection model

3.3. Preprocessing and Normalization

Efficient preprocessing is an essential phase in the proposed IDS model, guaranteeing that the data is formatted appropriately for analysis. The present research utilizes a multistage preprocessing approach, starting with feature selection and target variable mapping that simplifies the classification work. To rectify the class imbalance in the data set, SMOTE was utilized to produce synthetic data for minority classes, resulting in a balanced dataset. The first stage entails delineating the features and the target variable. The target variable is subsequently categorized into one benign class and seven overarching classes of attack, therefore mitigating the complexity of the problem.

SMOTE is employed on the data set to mitigate the issue of class imbalance. SMOTE functions by generating synthetic instances within minority classes, rectifying class distribution imbalance. Although SMOTE effectively mitigates class imbalance by generating supplementary samples for minority categories, it possesses inherent limitations. The synthetic data are validated by analyzing the class distribution and confirming that the created samples correspond with the statistical properties of the actual data. Nonetheless, the application of SMOTE can continue to impact the model's efficacy in highly dynamic practical IoT contexts [28].

Following SMOTE resampling, the features are normalized utilizing the StandardScaler from the sklearn library, which is also called Z-score normalization. This normalization guarantees that all features have a standard deviation of 1 and a mean of 0. This normalization process is essential for DL algorithms that are subject to changes in input data size.

The normalization procedure utilized in the research was defined by Equation (1). The Z-score normalization procedure constrains numerical values to the interval 0–1.

$$o' = \frac{o-\mu}{\sigma} \tag{1}$$

In this context, o represents the real value, $'$ denotes the standardized value, while σ and μ signify the standard deviation and mean values, respectively [29]. The subsequent step entails encoding a target variable with LabelEncoder. Subsequently, one-hot encoding is utilized to transform the categorical labels into the required format. The dataset is divided into training and testing sets, with 80% assigned for training and 20% assigned to testing.

3.4. Binary SSA-based Feature Selection

The Salp belongs to the Salpidae family, exhibiting a structure and behaviour comparable to that of jellyfish. SSA is a swarm intelligence system that emulates the behaviour of salps in the ocean.

Typically, salps exist in aggregations, forming structures known as salp chains. The main Salp was also designated as the leader, while the other salps were classified as followers. The position of the leader salps could be modified utilizing the computational model of SSA, as stated in the subsequent Equation (2).

$$z_i^l = \begin{cases} f_i + a_1\big((u_i - l_i)a_2 + l_i\big)\ a_3 \geq 0.5 \\ f_i - a_1\big((u_i - l_i)a_2 + l_i\big)\ a_3 \leq 0.5 \end{cases} \tag{2}$$

Here, z_i^l denotes the leader's position, while f_i Represents the i^{th} dimension of the food source's origin. The input variable a_1 plays a crucial role in SSA, gradually decreasing over the iteration in Equation (3) to facilitate greater exploration in the initial phases of the optimisation process and thus enhance exploitation in the following process. The terms l_i and u_i Signify the lower and upper bounds of the i^{th} dimension. The values a_2 and a_3 represent the random distribution that persists within the range [0, 1]. It guides the following update positions in the i^{th} dimension from -∞ to ∞, thus imposing the size of step [30].

$$a_1 = 2e^{-(4k/M)^2} \tag{3}$$

Here, k represents the current iteration, and M is the maximum number of iterations. The relevant positions are revised using the expression provided in Equation (4).

$$z_l^j = \frac{1}{2}\left(z_l^j - z_l^{j-1}\right) \tag{4}$$

Here, $j \geq 2$, and z_l^j Represents the location of the j^{th} affiliate in the i^{th} dimension.

The traditional SSA was employed to alleviate periodic optimization challenges. The binary metaheuristic algorithm is employed to address feature selection issues. According to bSSA, the salps are permitted to navigate within specific parameters defined by the values of zero and one. The primary method that affected the conversion process was the implementation of a transfer function. This function was employed to verify the likelihood of enhancing a component in the best possible solution as either one or zero. The sigmoidal transfer function was utilized to convert the continuous functions into an individual function, as articulated in Equation (5).

$$T\left(z_i^j(t)\right) = \frac{1}{1+\exp -z_i^j(t)} \tag{5}$$

Here, z_j^m Represents the j^{th} factor in the optimal value z within the j^{th} dimension, whereas t signifies the next iteration. The pertinent features could be chosen from the subset of features in accordance with Equation (6).

$$z_j^m(t+1) = \begin{cases} 1 \quad rand \geq T\left(z_j^m(t+1)\right) \\ 0 \quad rand \leq T\left(z_j^m(t+1)\right) \end{cases} \tag{6}$$

Here, z_j^m Denotes the j^{th} component of the optimum vector z in the j^{th} dimension.

Feature Selection (FS) aims to enhance accuracy while reducing the error rate and quantity of selected features. The proposed bSSA employs the objective function outlined in Equation (7) for the selection of the optimal feature set. The Fitness Function (FF) specified in Equation (7) is optimized throughout the selection of pertinent features.

$$FF = \alpha_1\gamma_x(D) + \beta_1\frac{|S-F_s|}{|S|} \tag{7}$$

Here, $\gamma_x(D)$ Denotes the accuracy associated with the decision D. The variable F_s Denotes the chosen features, S signifies the total number of features, $\alpha_1 \in [0,1]$ and β_1 are constants indicating the significance of classification accuracy and the length of the subset, respectively. The total of α_1 and β_1 must equal 1 $(\alpha_1 + \beta_1 = 1)$.

Equation (7) can be reformulated as a minimization problem utilizing error values and the ratio of selected features. The error-reducing fitness function is outlined in Equation (8).

$$FF = \alpha_1 E_R(D) + \beta_1\frac{|F_s|}{|S|} \tag{8}$$

Here, $E_R(D)$ Denotes the error rate [31].

A total of 15 optimal features have been selected from the CIC-IoT-2023 dataset using the BSSA approach. The features selected are ack_flag_number, fin_flag_number, header_length, https, icmp, magnitude, min, psh_flag_number, protocol, rst_count, rst_flag_number, syn_flag_number, tcp, udp, and variance.

3.5. Bidirectional Transformer Model

This research utilized the BiT approach for intrusion detection based on the selected features from the BD. This approach identifies and classifies attacks by analyzing BD IoT network traffic. To attain this objective, the BiT technique assimilates both historical and prospective information for enhanced detection accuracy. The architecture of the model is illustrated in Figure 4 [32].

The BiT model employs the mechanism of self-attention to encode the input sequence of data. For the input sequences $X = [x_1, x_2, \dots, x_T]$, where $x_t \in R^d$ Denotes the features that were input at time step t, the model initially produces three critical matrices via linear transformation: the query matrix Q, the key matrix K, and the value matrix V. The computational equations for these three matrix structures are as follows:

$$Q_t = XW_Q, K_t = XW_K, V_t = XW_V \tag{9}$$

In this Equation (9), $W_Q, W_K, W_V \in R^{d \times d_k}$ represents the acquired weight matrix and d_k Denotes the number of dimensions of every query and key.

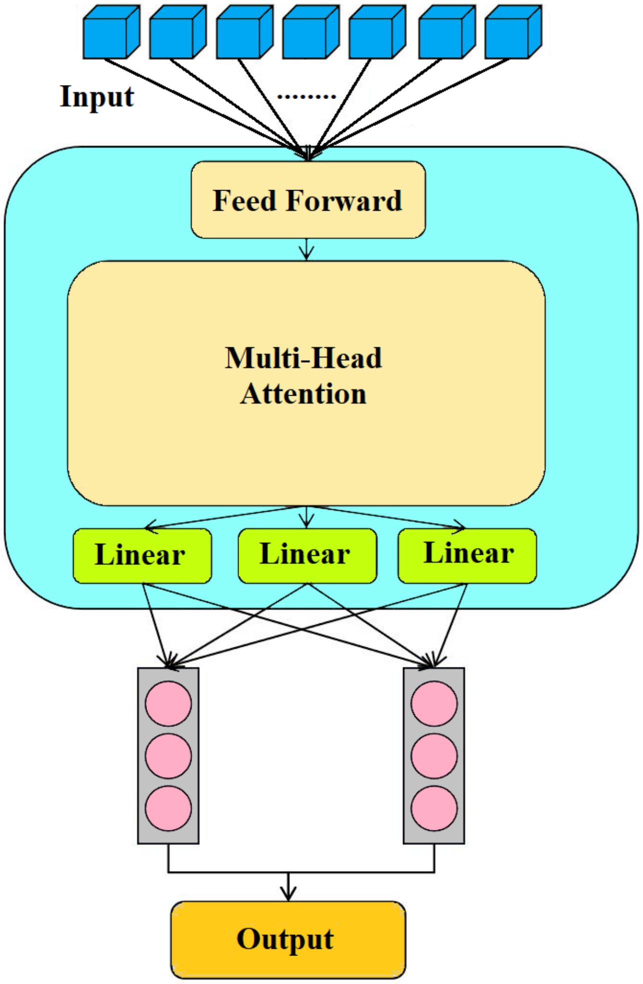

Fig. 4 Architecture of BiT Model

The correlation among several time steps is represented by computing the self-attention weight matrix. A_t Using Equation (10):

$$A_t = Softmax\left(\frac{Q_t K_t^T}{\sqrt{d_k}}\right) \tag{10}$$

The representation of every position in the sequence of inputs is derived by performing a weighted summation of the value matrix V, resulting in the output representation for each time step as given in Equation (11).

$$Z_t = A_t V_t \tag{11}$$

In the BiT, two channels of self-attention computation are employed: one is forward, from x_1 to x_T, and the second is reverse, from x_T to x_1. The model could simultaneously leverage prior attack patterns and predict potential future threats through this bidirectional exchange of data, enhancing its comprehension of the dynamic intrusion behaviours in IoT networks. To obtain the final representation, the model combined both the reverse and forward outputs as represented in Equation (12).

$$Z_t = [Z_t^{forward}, Z_t^{backward}] \tag{12}$$

Subsequently, the research enhances the model's expressiveness by layering several Transformer encoding layers. Every layer comprises a multi-head self-attention mechanism and a Feedforward Neural Network (FFNN).

In multi-head self-attention, distinct associations are acquired through numerous independent heads of attention h, and the last representation for every step is derived by combining the outputs from these heads and applying linear transformations.

$$Z_t = Concat(\sum_{h=1}^{H} A_t^h V_t^h) W_o \tag{13}$$

In this Equation (13), W_o denotes the learnt output weight matrix, whereas A_t^h and V_t^h signify the attention weight and value matrices of the h-th head, respectively, and H indicates the total number of attention heads. A FFNN is employed to execute a nonlinear change on the output of every layer, thereby enhancing the model's representational capacity. The computational procedure of the FFNN is delineated as given in Equation (14):

$$FFNN(Z_t) = ReLU(Z_t W_1 + b_1) W_2 + b_2 \tag{14}$$

Here, b_1, b_2 and W_1, W_2 represent the acquired biases and weights. In the end, following multiple layers of encoding, the model obtains a hidden state series $Z = [Z_1, Z_2, \dots, Z_T]$ That encompasses temporal information. To detect the intrusion, the work subsequently input this hidden state sequence into a model using linear Regression to estimate the potential threat level of the network data. The research computes the attack assessment value. y_t' Utilizing the subsequent formula in Equation (15):

$$y_t' = W_r Z_t + b_r \tag{15}$$

Here, $W_r \in R^{d \times 1}$ represents the learnt weight, $W_r \in R$ denotes the bias term, and y_t' signifies the attack detection value at the time step t. To optimize the model parameters, the research employs Mean Square Error (MSE) as the loss function. Given a true attack sequence $y = [y_1, y_2, \dots, y_T]$, the loss function L is articulated as in Equation (16):

$$L = \frac{1}{T} \sum_{t=1}^{T} (y_t - y_t')^2 \tag{16}$$

The model utilizes the back-propagation approach to adjust the parameters $W_Q, W_K, W_V, W_O, W_1, W_2, W_r$ Moreover, the bias in the model aims at minimizing the loss function. At the end, post-training, the model can detect intrusion attempts, allowing security measures to proactively address threats and mitigate the dangers associated with advancing intrusions in extensive IoT environments.

This method effectively captures nonlinear relationships and temporal dependencies in large-scale IoT network traffic data by utilizing a BiT architecture, integrated with a self-attention mechanism and an FFNN, thereby improving the accuracy of intrusion detection in BD environments [33]. The pseudocode of the developed IDS model is presented below.

```
Initialization
Load CIC-IoT-2023 dataset into Spark DataFrame
Drop irrelevant columns
Handle missing/null values
Encode categorical features
Map target variable labels to integers
Normalize features using Z-score normalization (StandardScaler in Spark)
Balance data using oversampling (SMOTE)
Split the dataset into training/testing sets
Feature Selection using BSSA
Initialize the salp population and control parameters
For each iteration:
    Evaluate the fitness of each Salp
    Update the positions of salps
    Select the feature subset with the best fitness score
End For
Select the best feature subset from BSSA
Initialize the Bi-directional Transformer model
Define the input layer based on selected features
Configure attention layers, positional encoding, and the classification head
Compile the model with optimizer
Train the model on training data
Predict on test data
Compute evaluation metrics
End
```

Table 3. Hyperparameter tuning of the model

Hyperparameter	Value
BSSA Population Size	50
No. of iterations of BSSA	1000
No. of Search Agents of BSSA	8
Search Domain of BSSA	[0, 1]
Learning Rate	0.001
Batch Size	32
Epochs	100
Dropout Rate	0.2
Hidden Size	256
Activation Function	ReLU
Optimizer	Adam

Table 3 shows the hyperparameter values set for the developed research model. The hyperparameter tuning ensures effective learning and model convergence while minimizing overfitting and optimizing computational efficiency for BD in Spark-based environments.

4. Experimentation Analysis

4.1. Experiment Setup

This section highlights the experiments conducted on the extensive CIC-IoT-2023 dataset. The study employed the PySpark tool, which facilitates programming with Python on the Apache Spark BD platform within the Google Colab environment. The proposed system is implemented using PySpark. It is a library that connects Python with Apache Spark. All testing was conducted on Windows 10 64-bit, utilizing a Core i7 processor operating at 2.70GHz, 16 GB of RAM, and the programming language, Python. The dataset is divided into testing and training halves of 20% and 80%.

4.2. Result Metrics

The research assessed the efficiency of the binary and multiclass classification models. The parameters of Accuracy, Precision, Detection Rate, and F1-score were computed to assess performance. The following are the formulas of these parameters presented in Equations (17) to (20).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{17}$$

Accuracy denotes the level of correlation or similarity between the predicted results and the true values within the dataset.

$$Precision = \frac{TP}{TP+FP} \tag{18}$$

Precision denotes the ratio of correctly identified positive instances to the total predicted positive cases. It is a key metric in the assessment of models.

$$Detection\ Rate = \frac{TP}{TP+FN} \tag{19}$$

Recall or Detection rate is a quantitative metric that assesses a model’s ability to identify all relevant occurrences accurately. These occurrences include the target class within a specified data set. A superior recall score indicates that the model can accurately identify a larger proportion of the dataset’s true positive samples. This demonstrates superior efficacy in identifying the target class.

$$F1score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{20}$$

The F1 score is an efficiency statistic that evaluates precision as well as recall to assess a model's accuracy. An elevated F1 score signifies improved overall effectiveness in reliably identifying positive samples. It is a significant indicator for evaluating the model's capacity to recognize the target class. The calculation employed the harmonic mean of recall and precision [11-26].

4.3. Performance Evaluation

This section discusses the results of the developed research model evaluated using the discussed performance indicators. Based on this BD framework, the results of the developed model are assessed in terms of both binary and multiclass classification. The results of the multiclass classification are compared with the current models for proper validation.

Table 4. Binary classification results of the research model

Metric (%)	Training	Testing
Accuracy	99.69	99.26
Detection Rate	99.55	99.13
Precision	99.72	99.18
F1-score	99.52	99.21

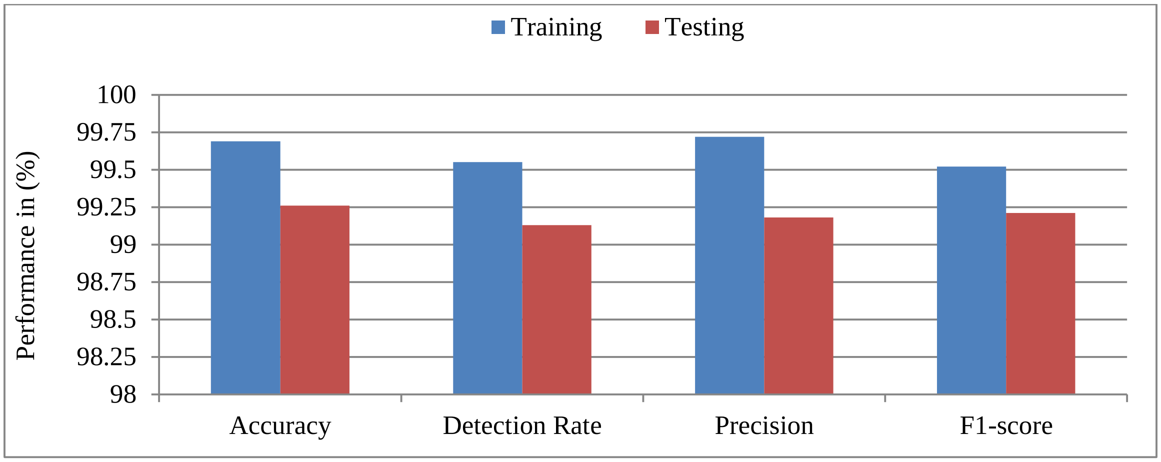

Fig. 5 Graphical illustration of BSSA-BiT model’s binary classification results

Table 4 presents the binary classification results of the developed IDS model. The binary classification results indicate that the model detects and classifies the attacks as either present or not. The name itself makes it clear that the attack represents 1 and 0 as normal. The results are individually evaluated for the training and test sets and compared. The accuracy of the developed model in training is 99.69% and 99.26% in testing. This binary accuracy performance highlights that the model has the ability to classify the attack accurately with a high generalization. The detection rate of the BSSA-BiT model in training was 99.55% and 99.13% in testing. This reflects the developed model’s capability to detect positive occurrences of the attack classes. The precision rate of the BSSA-BiT model in training was 99.72% and 99.18% in testing. This demonstrates that the model has the ability to reduce the false positives and has high reliability in predicting positive classes. Finally, the F1-score of the BSSA-BiT model attained an F1-score of 99.52% in training and 99.21% in testing. This highlights that the model has a balanced performance between precision and detection rate. Moreover, the model not only identifies the positive class, but it also effectively avoids misclassification. Thus, these binary classification results highlight that the BSSA-BiT model can perform robust and efficient intrusion detection. The binary classification results are plotted in Figure 5.

Table 5. Multiclass classification results of the research model

Classes	Accuracy	DR	Precision	F1-score
Benign	99.48	99.55	99.33	99.44
DDoS	99.21	98.97	99.50	99.23
Brute Force	98.89	98.65	98.78	98.71
Spoofing	98.35	97.90	98.20	98.05
DoS	98.76	98.10	99.00	98.55
Recon	99.05	98.93	98.85	98.89
Web-based	97.82	97.55	98.70	97.62
Mirai	99.30	99.10	99.22	99.16

Table 5 presents the multiclass classification results of the developed BSSA-BiT model. The multiclass classification results indicate that the model detects and classifies every attack in the data. The applied BD CIC-IoT-2023 dataset has a total of eight classes, in which seven are attacks and one is benign. The model attained high accuracy scores for all the classes in a consistent manner, ranging from 97.82% to 99.48%. The dominating attack classes like DDoS, DoS, and Mirai are detected with better accuracy scores. This indicates that the BSSA-BiT model has a strong generalization and effective adaptability. The detection rate of the model is similarly high for all the classes, ranging from 97.55% to 99.55%.

This detection rate performance highlights that the BSSA-BiT model can detect most of the occurrences accurately for each class. The precision rate of the model has been above 98% for each class, ranging from 98.70% to 99.33%. The model highlights that it can reduce false positives, particularly for DDoS and DoS attacks. The f1-score of the model ranges between 97.62% to 99.44%. Some slight variations exist in the performance for spoofing and web-based attacks. However, the overall performance highlights that the model is efficient and highly capable of detecting and classifying intrusions. Figure 6 depicts the graphical chart of the research model’s multiclass classification results.

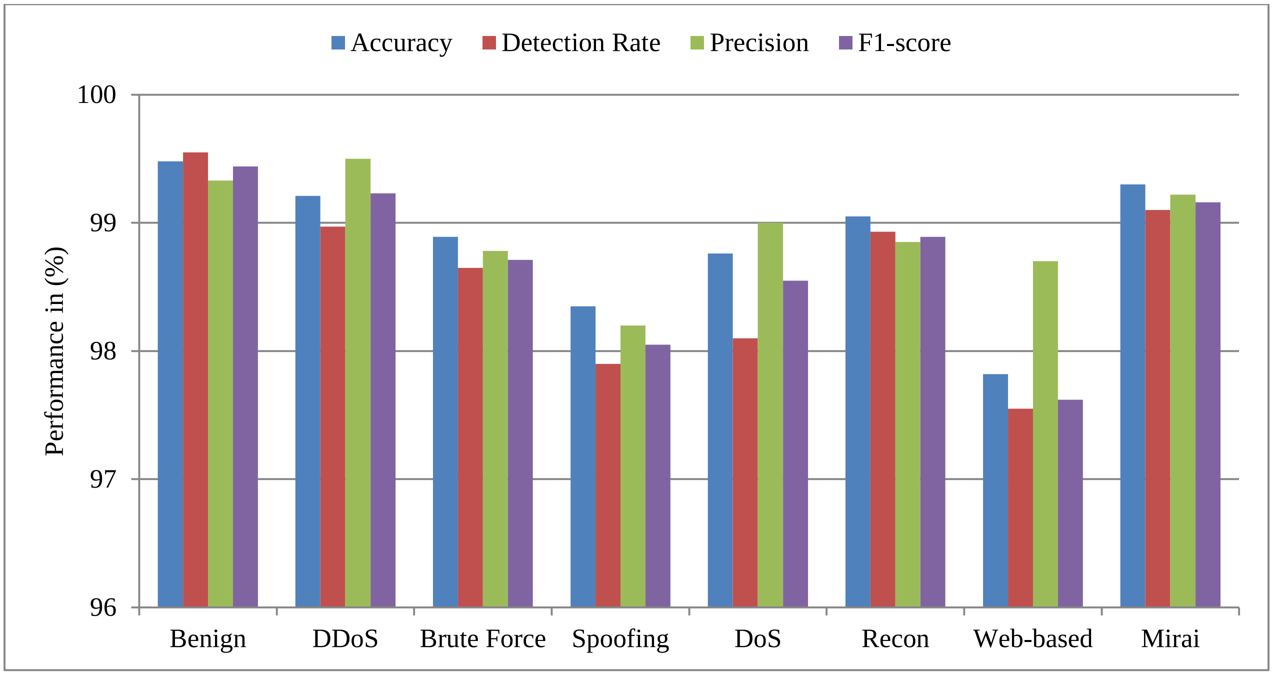

Fig. 6 Graphical illustration of BSSA-BiT model’s multiclass classification results

Table 6. Performance comparison with current models

Models	Accuracy	Detection Rate	Precision	F1-score
EHO-FDMM [11]	98.00	97.30	NA	NA
CNN-WDLSTM [13]	97.17	97.00	97.00	97.00
BiLSTM-CNN [14]	93.35	91.87	91.64	92.25
DL-SVM [15]	90.20	90.30	90.30	90.30
RF [17]	97.80	NA	NA	NA
EMBAM [18]	97.84	98.18	97.76	96.80
SSWLOFCC [19]	96.51	NA	NA	NA
DCGAN [22]	95.22	95.42	98.82	97.09
RF+XGB [23]	97.00	NA	NA	NA
TabNet [24]	96.00	92.00	94.00	95.00
MLP [25]	97.67	97.98	96.42	97.68
DL-SIS [26]	97.18	97.41	96.17	98.31
Proposed Model	98.85	98.59	98.94	98.70

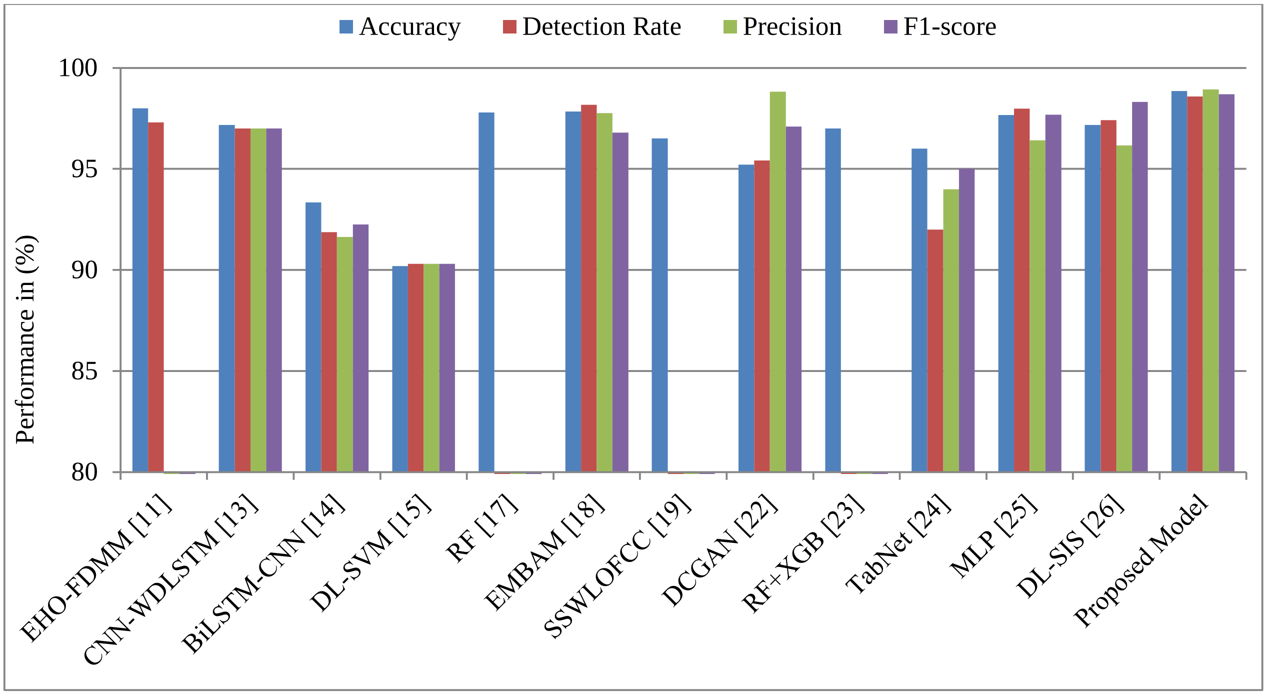

Fig. 7 Graphical illustration of results comparison

Table 6 presents an overall performance analysis comparison of the developed BSSA-BiT model's results with the other current models discussed in this research from the related works section. It is clearly demonstrated that the developed IDS model has attained high performance in all the evaluation metrics and outperformed all the current models. The average results of the research model from multiclass classification are applied for this comparison. The research model BSSA-BiT has achieved 98.85% accuracy, 98.59% detection rate, 98.94% precision, and 98.70% F1-score. In this comparison, models like EHO-FDMM, EMBAM, and MLP have results that are close to accuracy compared to the research model. Notably, the DCGAN model has attained very close precision score of 98.82%, but it falls short in accuracy and other metrics. Models like CNN-WDLSTM, BiLSTM-CNN, and DL-SVM have low-level performances with accuracies ranging from 90.20% to 97.17%. Models like EMBAM and RF+XGB also performed well but failed to match the performance of the developed BSSA-BiT model. The TabNet and DL-SIS models have also produced better results in F1-scores, but lag in detection rate and precision. Overall, the BSSA-BiT model's consistent performance and results outperformed the current models in this research on big-data-based intrusion detection. Figure 7 depicts the graphical chart of the research model's results compared with the current models.

This research has several advantages, including the fact that the model is highly accurate and effective in handling BD of IoT environments by developing a model that implements BSSA for feature selection and BiT for attack classification. Using Apache Spark in this research helped to improve the model's scalability and processing performances, which makes the BSSA-BiT model appropriate for BD-based intrusion detection systems. Moreover, the developed model demonstrates superior performance in the binary and multiclass classification results compared to the current models discussed in this research. However, there are a few limitations to this research model. The model has increased complexity due to its architecture. It led to higher computational costs and longer training.

5. Conclusion

This research proposed a BD-based IoT intrusion detection model using the DL algorithm with Apache Spark for attack detection and classification. The developed research model utilized the BSSA technique for feature selection and the BiT method for attack classification. The CIC-IoT-23 BD dataset was used to train and evaluate the model. The data was preprocessed with data cleaning, normalization, oversampling, and encoding. The binary variant of the BSSA technique selects the optimal features.

This feature selection process helped the classifier for accurate attack detection, minimizing dimensionality, and enhancing learning efficiency. The selected features were applied to train the BiT model. This DL architecture was capable of capturing contextual relations in sequential data and was optimal for complex intrusion detection tasks. After training, the model was evaluated using common evaluation metrics to assess its performance in detecting different attack types. The BSSA-BiT model attained 98.85% accuracy, 98.59% detection rate, 98.94% precision, and 98.70% F1-score in multiclass classification, and 99.26% accuracy, 99.13% detection rate, 99.18% precision, and 99.21% in binary classification. Compared to the other models, the BSSA-BiT model's consistent performance and results outperformed the current models of big-data-based intrusion detection in this research. In future, the developed research model can be extended to integrate the federated learning algorithm for data privacy in IoT. Additionally, an explainable AI technique can be used to improve the model interpretability, which could make the comprehension of decision-making simpler. Other BD tools like Apache Flink can be used for real-time data streaming and evaluation.

Acknowledgments

The authors would like to thank the Sona College of Arts and Science, Salem, and Thiruvalluvar Government Arts College, Rasipuram, for their support in completing this project.

References

[1] Afzal Badshah et al., "Big Data Application: Overviews, Challenge and Future," *Artificial Intelligence Reviews*, vol. 57, pp. 1-49, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Faheem Ullah, and Muhammad Ali Babar, "Architectural Tactic for Big Data Cybersecurity Analytic System: A Review," *Journal of System and Software*, vol. 151, pp. 81-118, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] Srikanth Thudumu et al., "A Comprehensive Survey of Anomaly Detection Techniques for High Dimensional Big Data," *Journal of Big Data*, vol. 7, pp. 1-30, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mohammad Shahnawaz, and Manish Kumar, "A Comprehensive Survey on Big Data Analytic: Characteristics, Tool and Technique," *ACM Computing Surveys*, vol. 57, no. 8, pp. 1-33, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] Isaac Kofi Nti et al., "A Mini-Reviews of Machine Learning in Big Data Analytic: Application, Challenge, and Prospect," *Big Data Mining and Analytics*, vol. 5, no. 2, pp. 81-97, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Lidong Wang, and Randy Jones, "Big Data Analytics in Cyber Security: Network Traffic and Attacks," *Journal of Computers Information System*, vol. 61, no. 5, pp. 410-417, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Nour Moustafa, *A Systemic IoT-Fog-Cloud Architecture for Big-Data Analytics and Cyber Security Systems: A Review of Fog Computing*, 1st ed., Secure Edge Computing, CRC Press, pp. 1-10, 2021. [Google Scholar] [Publisher Link]

[8] Zahedi Azam, Motaharul Islam, and Mohammad Nurul Huda, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis through Decision Tree," *IEEE Access*, vol. 11, pp. 80348-80391, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Seshagirirao Lekkala, Raghavaiah Avula, and Priyanka Gurijala, "Big Data and AI/ML in Threats Detections: A New Era of Cybersecurity," *Journal of Artificial Intelligences and Big Data*, vol. 2, no. 1, pp. 32-48, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Duan Dai, and Sahar Boroomand, "A Review of Artificial Intelligence to Enhance the Security of Big Data Systems: State-of-Art, Methodologies, Applications, and Challenges," *Archives of Computational Method in Engineering*, vol. 29, no. 2, pp. 1291-1309, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] V. Suresh Kumar, "A Big Data Analytical Framework for Intrusion Detection Based On Novel Elephant Herding Optimized Finite Dirichlet Mixture Models," *International Journal of Data Informatics and Intelligent Computing*, vol. 2, no. 2, pp. 11-20, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Noha Hussen et al., "A Full Streaming Big Data Framework for Cybersecurity Based on Optimized Deep Learning Algorithms," *IEEE Access*, vol. 11, pp. 65675-65688, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Mohammad Mehedi Hassan et al., "A Hybrid Deep Learning Model for Efficient Intrusions Detections in Big Data Environments," *Information Science*, vol. 513, pp. 386-396, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Hong Wang, "A Network Intrusion Security Detection Method Using BiLSTM-CNN in Big Data Environment," *Journal of Information Processing System*, vol. 19, no. 5, pp. 688-701, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Soosan Naderi Mighan, and Mohsen Kahani, "A Novel Scalable Intrusions Detections System based on Deep Learning," *International Journals of Information Security*, vol. 20, no. 3, pp. 387-403, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Mohamed Abushwereb et al., "An Accurate IoT Intrusions Detections Framework using Apache Sparks," *arXiv preprint*, pp. 1-15, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Selvam Saravanan, and Uma Maheswari Balasubramanian, "An Adaptive Scalable Data Pipelines for Multiclass Attacks Classifications in Large-Scale IOT Network," *Big Data Mining and Analytics*, vol. 7, no. 2, pp. 500-511, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Abdallah A. Alhabshy, Bashar I. Hameed, and Kamal Abdelraouf Eldahshan, "An Ameliorated Multiattack Network Anomaly Detection in Distributed Big Data System-Based Enhanced Stacking Multiple Binary Classifiers," *IEEE Access*, vol. 10, pp. 52724-52743, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Riyaz Ahamed Ariyaluran Habeeb et al., "Clustering-Based Real-Time Anomaly Detection—A Breakthrough in Big Data Technologies," *Transaction on Emerging Telecommunication Technologies*, vol. 33, no. 8, pp. 1-27, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Rong Chen, "Design and Protection Strategy of Distributed Intrusions Detections Systems in Big Data Environments," *Computational Intelligences and Neurosciences*, vol. 2022, no. 1, pp. 1-7, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Guo Yunhong, and Tang Guoping, "Intelligent Analysis and Dynamic Security of Networks Traffics in Contexts of Big Data," *Journal of Cyber Security and Mobility*, vol. 13, no. 5, pp. 823-842, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Yixuan Wu et al., "Intelligent Intrusion Detection for Internet of Things Security: A Deep Convolutional Generative Adversarial Network-Enabled Approach," *IEEE Internet of Things Journals*, vol. 10, no. 4, pp. 3094-3106, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Farah Jemili, Rahma Meddeb, and Ouajdi Korbaa, "Intrusions Detections Based on Ensembled Learning for Big Data Classifications," *Cluster Computing*, vol. 27, pp. 3771-3798, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] Bhukya Madhu et al., "Intrusion Detection Models for IOT Networks via Deep Learning Approaches," *Measurement: Sensors*, vol. 25, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[25] Murat Dener, Gökçe Ok, and Abdullah Orman, "Malwares Detections using Memory Analysis Data in Big Data Environments," *Applied Sciences*, vol. 12, no. 17, pp. 1-21, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Feilu Hang et al., "Research on the Application of Network Security Defence in Database Security Services based on Deep Learning Integrated with Big Data Analytics," *International Journal of Intelligent Network*, vol. 5, pp. 101-109, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Viet Anh Phan, Jan Jerabek, and Lukas Malina, "Comparison of Multiple Feature Selection Techniques for Machine Learning-Based Detection of IoT Attacks," *Proceedings of the 19th International Conference on Availability, Reliability and Security*, Vienna Austria, pp. 1-10, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[28] Seyed Mohammad Hadi Mirsadeghi et al., "Learning From Few Cyber-Attacks: Addressing the Class Imbalance Problem in Machine Learning-Based Intrusion Detection in Software-Defined Networking," *IEEE Access*, vol. 11, pp. 140428-140442, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[29] Dasheng Chen et al., "Identification of Network Traffic Intrusion Using Decision Tree," *Journal of Sensors*, vol. 2023, no. 1, pp. 1-9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[30] S. Jayachitra et al., "An Efficient Ranking Based Binary Salp Swarm Optimization for Feature Selection in High Dimensional Datasets," *Measurement: Sensors*, vol. 35, pp. 1-5, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[31] Sayar Singh Shekhawat et al., "bSSA: Binary Salp Swarm Algorithm with Hybrid Data Transformation for Feature Selection," *IEEE Access*, vol. 9, pp. 14867-14882, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[32] Murat Tezgider, Beytullah Yildiz, and Galip Aydin, "Text Classification using Improved Bidirectional Transformer," *Concurrency and Computations: Practice and Experience*, vol. 34, no. 9, pp. 1-12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[33] Hany El-Ghaish, and Emadeldeen Eldele, "ECGTransForm: Empowering Adaptive ECG Arrhythmias Classifications Frameworks with Bidirectional Transformers," *Biomedical Signals Processing and Control*, vol. 89, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]