Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.084282

ARTICLE

A Scalable Deep Learning Framework for Real-Time Cyber Threat Detection in Big Data Security Analytics

Salman Khan* **and Mai Alzamel***

Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

*Corresponding Authors: Salman Khan. Email: snawab@ksu.edu.sa; Mai Alzamel. Email: malzamel@ksu.edu.sa

Received: 20 April 2026; Accepted: 27 May 2026; Published: 30 June 2026

ABSTRACT: Traditional threat detection has proven ineffective in large-scale, moving data in the era of ever-more complex adversarial techniques and interconnected systems. The challenge becomes even more complex when high-volume, unstructured data continuously streams from social media platforms, requiring them to process the data efficiently and intelligently to provide timely security insights. Considering the big data security, the present study presents a scalable deep-learning-based system for real-time cyber threat detection, which has been developed and validated especially for distributed big data processing environments. A hybrid embedding approach that combines Word2Vec and Iterated Dilated Convolutional Neural Networks (ID-CNNs) is created to extract complementary semantic and sequential linguistic patterns from tweet data. Next, the most informative features are selected using SHapley Additive exPlanations (SHAP) feature selection technique by discarding redundant and noisy signals, followed by a Deep Neural Network (DNN) classifier to classify threats in real-time. The framework is deployed and tested on two well-known big data platforms (Apache Hadoop, Apache Spark) with different-sized data sets and node configurations to measure detection accuracy as well as computational scalability. From the classification point of view, when it comes to accuracy, the proposed model gives a 99.18% accuracy rate and MCC of 0.984 which is better than all the baseline models. When it comes to scalability, Apache Spark has proven to be superior to Apache Hadoop in all configurations, with up to 5.10 times speedup on the biggest dataset, still maintaining classification accuracy, and dropping execution time from 53 s on a single node to 12 s with six nodes on 8M threads. The results validate that the proposed framework provides a powerful, highly accurate, and computationally efficient solution for real-time cyber threat intelligence in large-scale big data security analytics applications.

KEYWORDS: Spark; Hadoop; cyber threat intelligence; DNN; big data

1 Introduction

The current digital environments are defined by the ubiquity of social media in everyday communication, commerce, and institutional processes. These platforms act as high-throughput channels through which a variety of people, organizations, and interest groups interact and share information [1]. At the same time, their growth has expanded the space for criminals to operate, enabling them to engage in a wide range of illegal activities in these environments [2]. The most common methods include creating bogus user accounts, sending messages with embedded URLs, and spoofing legitimate organizations to trick users and trick them into revealing personal information. A lot of these attacks depend on the public data of the profile that has been gathered, such as the interactions, declared affiliations, and user preferences, as part of social engineering operations and pre-intrusion reconnaissance. The complexity of these threats and their frequency have made cybersecurity a strategic need for organisations in both public and private sectors. The

impact of cybercrime is now conservatively estimated to cost about 0.8% of global GDP, and is a pervasive and systemic problem [3]. In this context, the Security Operations Centre (SOC) needs to develop strong situational awareness, with constant access to intelligence that is timely, operationally relevant, and helps to defend critical digital infrastructure effectively. All these factors have made Cyber Threat Intelligence (CTI) a hot topic in the cybersecurity conversation. CTI is a formalized discipline that is concerned with the systematic collection, processing, and analysis of threat-relevant information and is intended to promote anticipatory and responsive defensive postures. CTI uses structured analytical frameworks to transform raw technical artefacts, such as IDS outputs, malware fingerprints, and phishing indicators, into intelligence that can be used to make decisions about protecting operations [4]. One of the core elements of CTI practice is Open Source Intelligence (OSINT) [5], which leverages the ocean of information generated by social media, community platforms, technical literature, and open-access blogs and facilitates the rapid identification and interpretation of threat signs. In this context, X (formerly Twitter) is an OSINT tool par excellence. It's a real-time, high-speed environment of information that is populated by cybersecurity experts, research organizations, and the public—a place where intelligence is constantly added about new vulnerabilities, breaches in use, and new tactics. Systematic gathering and analysis of this content stream can be used to identify and track cyber incidents over time, as well as reveal the tactics, techniques, and procedures (TTPs) of threat actors [6]. In this context, the development of a new and interesting, yet ongoing research project in cybersecurity—automated, high-fidelity pipelines for extracting actionable intelligence from open digital platforms—has become a promising direction. Against this backdrop, a new and interesting, yet ongoing research project in cybersecurity, that of automated, high-fidelity pipelines for extracting actionable intelligence from open digital platforms, has emerged as a promising direction [7].

Scholarly interest in exploiting X as an analytical resource for cybersecurity threat identification has grown substantially in recent years. Seminal contributions have laid the conceptual groundwork: Ruohonen et al. [8] emphasise the need to trace security vulnerabilities to their root causes, while Abdelhaq et al. [9] highlight the practical value of developing targeted remediation strategies once weaknesses are characterised. Extending these foundations, Sabottke et al. [10] harnessed X's Streaming API alongside an SVM classifier to track CVE references across the platform, generating early exploit warnings with a median lead time of 2 days. Complementing this work, Kergl et al. [11] devised a community-driven detection scheme that applied domain-specific keyword sets to surface emerging zero-day vulnerabilities as they propagated through the platform's information ecosystem. Subsequent work has increasingly turned to machine learning to formalise threat detection from social media streams [12–14]. Ref. [15] developed a centroid-based architecture capable of isolating security-pertinent content without reliance on CVE identifiers, attaining an F1 score of 64%—a result that simultaneously illustrates the potential and the residual limitations of reference-free detection. Ref. [16] constructed an SVM pipeline to separate genuine security alerts from benign discourse, recording 94% classification accuracy, though residual misclassification remains an acknowledged limitation. The field has subsequently advanced through the adoption of deep learning [17–20], with CNN and BiLSTM architectures [3] demonstrating strong real-time classification of security-relevant content across multiple threat categories. Further architectural refinements include a multitask learning paradigm that integrates IDCNN with BiLSTM [15] and an event detection method that combines locality-sensitive hashing with incremental clustering for continuous, dynamically updated threat monitoring [21]. Collectively, these contributions affirm X's considerable value as an early-warning intelligence platform in the cybersecurity domain. Yet meaningful limitations endure, particularly in deepening contextual comprehension of nuanced threat discourse and in curbing the false-positive rates that continue to constrain operational deployment.

Moreover, the continuous and permanent nature of data generation in large-scale environments creates significant storage and computational challenges, which massive parallel and distributed computing

platforms have proven effective in addressing [22]. Several distributed frameworks have been developed for large-scale data processing [23], including Flink [24], Apache Storm [25], Apache Samza [26], Hadoop MapReduce [27], and Apache Spark [28]. Among these, Apache Hadoop and Apache Spark are the most widely adopted for big data analytics. Hadoop parallelises computation across thousands of processing nodes and has established itself as a reliable platform for large-scale data analysis, though it is limited to offline batch processing [29]. Apache Spark offers a lighter alternative with the same parallelisation capability but adds in-memory processing, enabling real-time data examination and stream processing. Studies in the literature confirm that Spark consistently outperforms Hadoop MapReduce in computational efficiency [27], primarily because Spark retains intermediate data in memory rather than writing results back to disk after each Map and Reduce operation as Hadoop does. However, Spark's performance advantage diminishes when dataset sizes exceed available memory capacity [30].

To fill this gap, in this study, a scalable, machine learning-based real-time cyber threat detection system from social media data is proposed. Tweets are retrieved using the X API and filtered by cybersecurity-related keywords to make sure that the corpus will be relevant. A hybrid embedding strategy, combining Word2Vec and Iterated Dilated Convolutional Neural Networks (ID-CNNs), is devised to learn complementary semantic and sequential linguistic information from the tweet texts. Signal selection methods are then used to identify the most discriminative signals (e.g., SHapley Additive exPlanations (SHAP) method) while removing redundant and noisy dimensions. The enhanced feature set is then fed to a Deep Neural Network (DNN) classifier for real-time threat modelling and classification. The framework is tested at two levels: (1) The effectiveness of the model is tested on a benchmark dataset, and the average accuracy obtained by the proposed model is 99.18% with an MCC of 0.984, which is superior to all the baseline models; (2) The computational efficiency of the proposed model is tested using Apache Hadoop and Apache Spark, and Spark is found to have lesser time. The key contributions of this work are as follows:

1. A hybrid feature extraction strategy with feature selection applied to retain the most discriminative signals from cybersecurity-related social media text, achieving superior classification performance over existing state-of-the-art methods.
2. A parallel DNN classifier is deployed and evaluated on both Apache Hadoop and Apache Spark, providing a practical assessment of the proposed framework across two established big data processing paradigms.
3. A direct comparison between Apache Hadoop and Apache Spark shows that Spark achieves up to 19.38 times faster execution, with an average speedup of 17.79 across all configurations, while fully preserving classification accuracy, confirming its practical advantage for real-time, large-scale cyber threat detection.

2 Proposed Model

In this section, we introduce the design of the proposed model. The architecture of the proposed model is shown in Fig. 1, which includes several components discussed in detail below.

2.1 Feature Encoding Schemes

2.1.1 Word2Vec-Based Feature Extraction

To obtain distributed representations of words in the tweet corpus, Word2Vec was used. Unlike other methods like Bag-of-Words or TF-IDF, Word2Vec does not ignore the context of words; instead, it is based on the idea that words have semantic relationships as well as contextual relationships, which can be expressed by vector space mappings. Words sharing the same semantic or syntactic function are placed close together in

this space, adding greatly to the quality of the learned representations. Word2Vec serves as a shallow neural network that achieves dense low-dimensional embeddings of words, encodes their co-occurrence structure, and semantic proximity through a sliding window technique, which are both efficient and very effective for downstream classification, clustering, or sentiment tasks and are often highly efficient and effective [31].

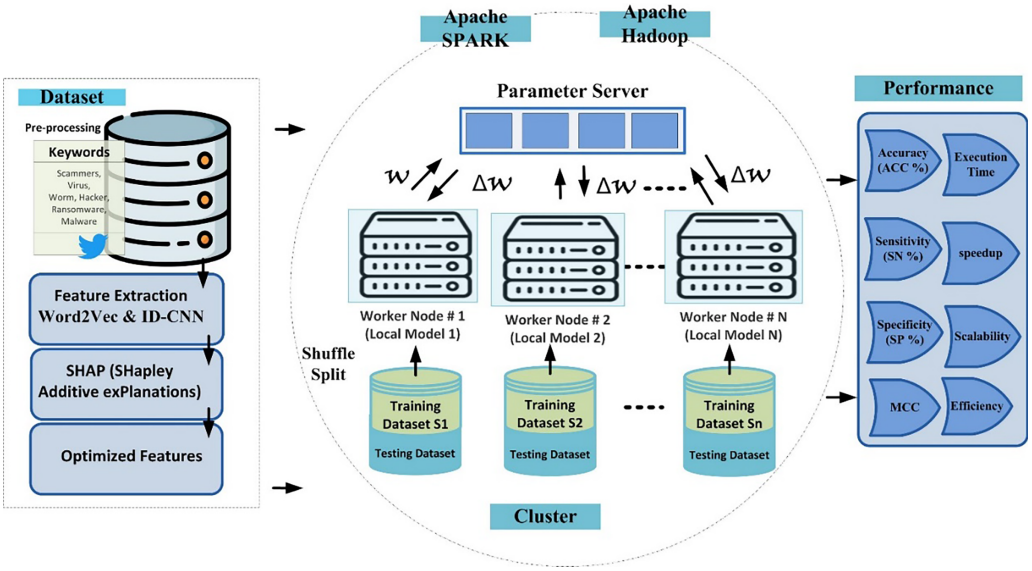

Figure 1: The proposed model architecture.

Two complementary training strategies are available in Word2Vec i.e., the Continuous Bag-of-Words (CBOW) and Skip-Gram models [32]. The two architectures are similar in that they both have three layers: input, hidden, and output—but have different goals: predictions. CBOW uses surrounding context tokens to predict a target word, providing computational efficiency and excellent performance for common words. In contrast, skip-gram predicts the context words within a window of a target word, showing the best generalisation on unusual words and larger corpora. Both variants were used in this work, and the resulting embeddings were joined together to form a feature vector. Each model generates 300 features per word, and the concatenation of the 300 features per word from each model results in 600 features per word. Global pooling was performed to obtain tweet-level representations by averaging all the word embeddings in each tweet, resulting in a single 600-dimensional vector that captures the semantic and contextual characteristics of a tweet.

2.1.2 Iterated Dilated Convolutional Neural Networks (ID-CNNs)

In the context of Natural Language Processing (NLP), Convolutional Neural Networks (CNNs) are adapted to process one-dimensional sequences of high-dimensional word embeddings [33]. Unlike standard convolutions, which are constrained by a fixed filter width w, this study utilizes Iterated Dilated Convolutional Neural Networks (ID-CNNs) to capture broader contextual patterns without parameter inflation. Formally, the dilated convolutional operator ct at position t is defined as

$$ct = \mathrm{W_c}.\left[x_{t-r\delta}; \ldots; x_{t+r\delta}\right] \tag{1}$$

where δ represents the dilation width. In our architecture, the ID-CNN module is composed of three dilated layers with progressive dilation rates of 1, 2, and 3. This configuration exponentially expands the receptive

field, allowing the model to aggregate multi-scale dependencies from surrounding tokens while maintaining the original sequence resolution. To enhance feature extraction from inherently noisy cybersecurity social media text, the dilation block is iterated four times. Each layer employs 128 filters to generate dense feature vectors, while the iterative structure serves as a mechanism for deep contextual encoding. By sharing parameters across these iterations, the model achieves a high-capacity representation of complex, multi-word threat indicators and domain-specific expressions, effectively balancing architectural depth with a robust defense against overfitting

2.2 Hybrid Features and Selection

Word2Vec representations and ID-CNN feature vectors are fused into a single hybrid embedding that is able to capture the semantic and sequential dimensions of the input text simultaneously. Word2Vec captures local and word-level semantic relationships in words, and ID-CNNs use a hierarchy of dilated convolutions to capture the sequential and local patterns in the context. The extracted vectors are concatenated into a single hybrid feature vector to extract the most discriminative information from the combined feature space and to boost the predictive strength. Specifically, the feature vector output by CBOW and Skip-Gram is concatenated to get a 600-dimensional Word2Vec representation, and then fused with an ID-CNN output of a 128-dimensional feature vector to get a 728-dimensional combined feature vector.

$$\mathrm{T}_{\mathrm{Threats}} = ID - CNN_{128} || Word2Vec_{600} \tag{2}$$

Quantifying the contribution of individual features to model behaviour presents a persistent challenge in machine learning, particularly given the opaque nature of many high-performing models. Identifying which features meaningfully drive predictions is essential for improving model performance, mitigating noise, and reducing computational overhead. To address potential redundancy, noise, and irrelevant dimensions in the high-dimensional hybrid feature space, SHAP was used for feature selection. Grounded in cooperative game theory, SHAP assigns importance values to individual features by quantifying their marginal contribution to model outputs, supporting transparent and principled interpretation. Eq. (2) provides the formal mathematical expression for this process, capturing the incremental effect of incorporating feature i across different feature subsets.

$$\mathrm{SHAP_i}\,(\mathrm{x}) = \varnothing_\mathrm{i} = \sum\nolimits_{\mathrm{s}\subseteq \mathrm{N}\{\mathrm{i}\}} \frac{|\mathrm{S}|\,(|\mathrm{N}| - |\mathrm{S}| - 1)}{|\mathrm{N}|}\,[\mathrm{f}\,(\mathrm{S}\cup\{\mathrm{i}\}) - \mathrm{f}(\mathrm{S})] \tag{3}$$

Fig. 2 visualises the top 10 SHAP features (i.e., from the selected 512 optimised features). Each row represents an individual feature, and each point encodes its value for a specific instance. Red points correspond to high feature values; blue points correspond to low feature values. The horizontal axis reflects SHAP importance scores, indicating the directional influence of each feature on predicted class probabilities. Features with positive SHAP values increase the predicted likelihood of cyber threat classification, while those with negative values elevate the probability of a non-threat prediction. Through this selection procedure, a refined 512-dimensional feature set was obtained, substantially reducing input complexity while preserving the most predictive signal.

2.3 Architecture of the Deep Neural Network

Deep neural networks (DNNs) are a class of machine learning architectures that model complex nonlinear functions through hierarchical feature transformations. A standard DNN comprises an input layer, one or more hidden layers, and an output layer [34]. The hidden layers are central to the network's capacity to discover latent patterns and structural regularities that are not directly observable from raw input

features. Increasing the depth of hidden layers enhances representational capacity and the ability to model intricate mappings. However, this comes at the cost of greater computational demand, increased training complexity, and a heightened risk of overfitting. A particularly valuable property of DNNs is their capacity for automatic feature learning, enabling the direct derivation of task-relevant representations from raw data without manual engineering—a capability especially advantageous for unstructured or unlabelled inputs. Their versatility has driven widespread adoption across fields including bioengineering, image recognition, audio processing, and natural language processing. The proposed DNN is trained on benchmark datasets, as depicted in Fig. 3. The architecture encompasses an input layer, an output layer, and five hidden layers, with inter-layer connectivity governed by the feature vectors defined in Eq. (4), consistent with established architectural conventions.

$$y_a = f\left(B_a + \sum_{b=1}^{m} x_b w_b^a\right) \tag{4}$$

where y_a denotes layer output, B_a bias term, w_b^a weight matrix, x_b input feature, and a nonlinear Tanh activation f function computable via Eq. (5).

$$f(i) = \frac{e^i}{1 + e^i} \tag{5}$$

The DNN hyperparameter configuration was determined through an exhaustive grid search over a systematically defined parameter space. Three stochastic parameters received particular attention: iteration count, activation function type, and learning rate. Grid search results indicated LR 0.01; the Tanh activation function was validated under 10-fold cross-validation, with peak classification accuracy as shown in Table 1.

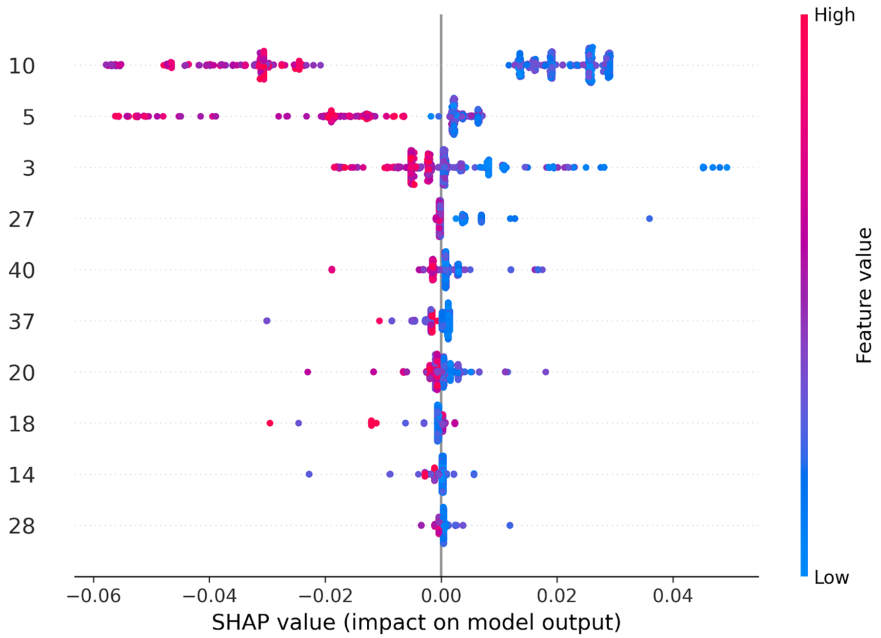

Figure 2: Selected features with SHAP.

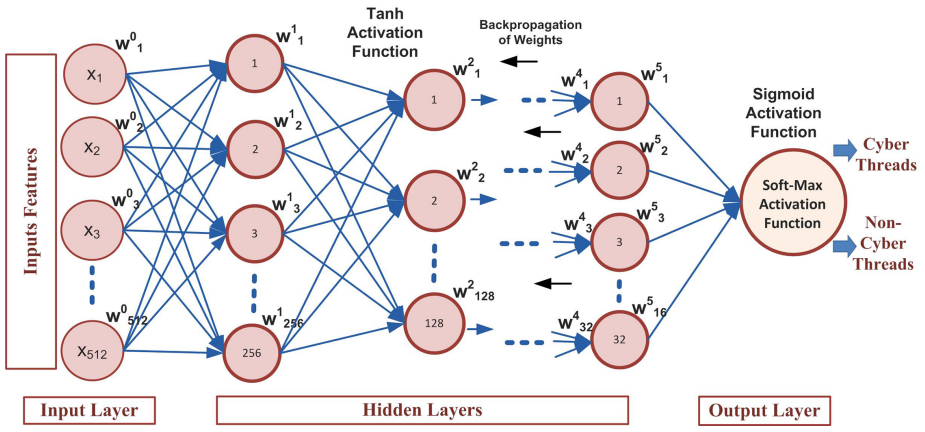

Figure 3: Deep neural network model architecture.

Table 1: Optimal hyperparameters of the proposed DNN.

Hyper Parameters	Optimal Values
Momentum	0.9
Optimizer	Adam
Input Layer	512
Hidden Neurons	256-128-64-32-16
Dropout	0.25
Activation Functions	Tanh and Sigmoid
Weight initialisation function	XAVIER function
Dense layers	5
Training Epoch	50
Batch Size	32
Regularisation L1	0.001
Learning rates	0.01

In this study, a parallel Deep Neural Network (DNN) is implemented and evaluated on both Apache Spark and Apache Hadoop to assess the scalability of the proposed cyber threat detection framework under big data processing conditions, as illustrated in Fig. 1. In the Spark-based implementation, large training data is partitioned into Resilient Distributed Datasets (RDDs) and distributed across a cluster of worker nodes, where a copy of the DNN model is trained simultaneously on each partition using both data and model parallelisation. Upon completion of training, each worker node returns its locally trained model to the master node, where an average parameter function aggregates the results into a single global model, which is then applied to distributed test partitions to produce the final classification output. To optimise the proposed model, we apply the backpropagation technique to each model deployed on the worker nodes. At each iteration, the DNN model alternates between minimising the loss function.

3 Experiment Result

3.1 Benchmark Dataset

The quality and representational characteristics of training data are foundational to the performance of any machine learning model. Publicly available benchmark datasets were selected to train and validate

the proposed model. The primary training corpus is the dataset curated by Khan et al. [35] and also by Syed Abbas Raza, available via Kaggle (https://www.kaggle.com/datasets/syedabbasraza/suspicious-tweets/data). To standardise the textual quality of the Kaggle dataset and eliminate noise sources, a structured five-stage preprocessing pipeline was implemented. First, custom regular expressions were applied to remove emoji characters, which carry inconsistent semantic interpretations across contexts and were excluded to reduce ambiguity in downstream analysis. Second, URLs and excessive punctuation were stripped from all text, as URLs provide no linguistically meaningful signal for threat classification, and excessive punctuation can disrupt tokenisation fidelity. Third, all tokens were converted to lowercase to enforce orthographic uniformity, preventing the model from treating identical words in different cases as semantically distinct tokens. Fourth, stop words were removed using the NLTK library [36], as these high-frequency function words contribute negligible discriminative information and unnecessarily inflate feature dimensionality. Fifth, the Snowball stemmer was applied to reduce words to their morphological roots, collapsing inflectional variants into a unified representation and enabling the model to recognise semantic equivalences across different surface forms. Upon completion of all five stages, the cleaned corpus retained 40,838 cyber threat instances and 7945 non-threat instances, ready for subsequent model training and evaluation.

3.2 Performance Evaluation

A comprehensive evaluation of deep learning models requires the use of multiple complementary performance metrics. Assessment typically commences with a confusion matrix, which decomposes classification outcomes into true positives T^+, false positives F^+, true negatives T^-, and false negatives F^- across training iterations. To thoroughly characterise the proposed model's performance, five standard metrics were computed: Accuracy (ACC), Specificity (SP), Sensitivity (SN), F1-score, and Matthews Correlation Coefficient (MCC). ACC quantifies the overall proportion of correct classifications. SP measures the true negative rate—the classifier's ability to identify non-threat instances correctly. SN captures the true positive rate, reflecting the proportion of actual threat instances correctly detected. The F1-score provides a harmonically balanced summary of precision and recall, accounting for both false positives and false negatives. MCC, also known as Matthews Correlation Coefficient, robustly evaluates binary classification performance under class imbalance [35]. These metrics are formally defined using the Chou notation as follows:

$$\mathrm{ACC} = \frac{T^+ + T^-}{T^+ + F^+ + T^- + F^-} \tag{6}$$

$$\mathrm{SP} = \frac{T^-}{F^+ + T^-} \tag{7}$$

$$\mathrm{SN} = \frac{T^+}{T^+ + F^-} \tag{8}$$

$$F1 - Score = \frac{2 * TP}{2 * TP + FP + FN} \tag{9}$$

$$\mathrm{MCC} = \frac{(T^- * T^+) - (F^- * F^+)}{\sqrt{(f^+ + T^+)(T^+ + F^-)(F^+ + T^-)(T^- + F^-)}} \tag{10}$$

3.3 Performance Analysis

This section presents a comparative evaluation of the proposed DNN model under a sequential processing approach. Feature representations derived from Word2Vec and ID-CNNs, evaluated individually and in hybrid combination, were assessed to determine the contribution of each component to overall classification performance. Table 2 summarises classification performance across all feature extraction configurations.

Table 2: Proposed model performance evaluation.

Method	ACC (%)	SP (%)	SN (%)	MCC
W2V	98.17	97.85	98.48	0.963
ID-CNNs	98.54	98.42	98.65	0.971
Hybrid Features	99.07	99.09	99.06	0.981
Hybrid Features-SHAP Based	99.18	99.16	99.20	0.984

From Table 2, Word2Vec alone achieved 98.17% accuracy and an MCC of 0.963. In comparison, ID-CNNs independently achieved 98.54% accuracy and an MCC of 0.971, confirming that sequential feature extraction provides a stronger representational foundation than semantic embeddings alone. Combining both into a hybrid feature vector further improved performance to 99.07% accuracy with an MCC of 0.981, demonstrating that the two representations capture complementary information that neither encodes individually. Following SHAP-based feature selection to eliminate redundant and noisy dimensions, accuracy improved to 99.18% with an MCC of 0.984, confirming that dimensionality refinement meaningfully enhances predictive efficiency without discarding discriminative content.

Fig. 4 presents the confusion matrices of the optimised hybrid feature configuration, providing granular insight into prediction behaviour across class boundaries. These results collectively demonstrate that the proposed hybrid feature extraction strategy, combined with SHAP-based selection, delivers superior classification performance relative to all individual feature configurations.

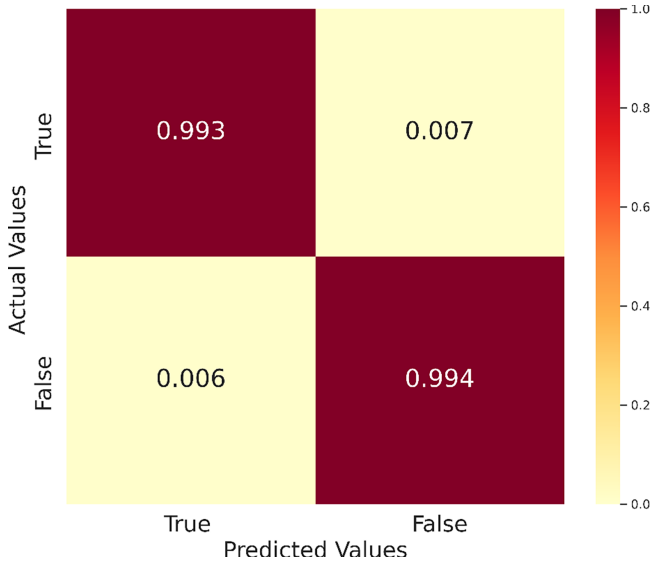

Figure 4: Confusion metrics of the proposed model using optimised hybrid features.

3.4 Other Learning Classifiers Comparison

To contextualise the proposed model's performance, it was benchmarked against four widely adopted machine learning classifiers: KNN, NB, SVM, and RF, all evaluated using the optimal hybrid feature representation. Table 3 reports classification outcomes.

Table 3: Performance comparison with other ML algorithms.

Methods	ACC (%)	SP (%)	SN (%)	MCC
Proposed Model	99.18	99.17	99.18	0.984
SVM	98.56	98.51	98.62	0.971
NB	98.47	98.41	98.53	0.969
KNN	98.38	98.48	98.29	0.968
RF	98.33	98.21	98.45	0.967

The proposed model attained the highest accuracy of 99.18%, surpassing all comparator classifiers. SVM ranked second-strongest, achieving 98.56% accuracy with an MCC of 0.971. Overall, the proposed model demonstrated consistent superiority over all competing classifiers, improving the average accuracy of 0.74%.

3.5 Comparison with Existing State-of-the-Art Models

The proposed framework was further evaluated against established benchmark approaches. Table 4 presents a comprehensive comparison across key performance indicators, including F1-score and recall. Among the baseline methods, Mahaini and Li [7] achieved an F1-score of 90%, while Alves et al. [13] and Dionisio et al. [20] recorded recall values of 90.00% and 94.00%, respectively. Alsodi et al. [18] established a demanding performance threshold of 99.00% F1-score. The proposed model surpasses all prior methods with an F1-score of 99.18%, demonstrating greater capacity to maintain an optimal precision-recall balance.

Table 4: Performance comparison of the proposed model with existing models.

Methods	Model Performance
Proposed Model	F1-score 99.18%
Alsodi et al. [18]	F1-score 99.00%
Dionisio et al. [20]	Recall 94.00%
Alves et al. [13]	Recall 90.00%
Mahaini and Li [7]	F1-score 90%

3.6 Apache Hadoop versus Apache Spark

Both frameworks were evaluated across dataset subsets of varying sizes—2, 4, 6, and 8M cyber threats. As shown in Fig. 5, Apache Spark steadily achieved greater accuracy than Hadoop across all dataset sizes. Accuracy improved incrementally with increasing sequence counts for both frameworks, though the margin in favor of Spark widened slightly at larger scales. These results confirm that both frameworks maintain reliable classification performance, with Spark demonstrating a marginal but consistent advantage.

Fig. 6 presents execution time comparisons between the two frameworks across varying dataset sizes. Spark dramatically outperformed Hadoop MapReduce in all configurations. Processing 2M sequences required 58.30 s under Hadoop compared to just 3.60 s under Spark. At 8M sequences, Hadoop required

232.60 s while Spark completed the task in 12 s. In terms of relative speedup, Spark was 16.19 times faster than Hadoop on the 2M dataset and 19.38 times faster on the 8M dataset, yielding an average speedup of 17.79 times across all configurations. Spark's superior performance is primarily attributable to its reduced I/O latency, efficient utilisation of computational resources, and in-memory data processing architecture.

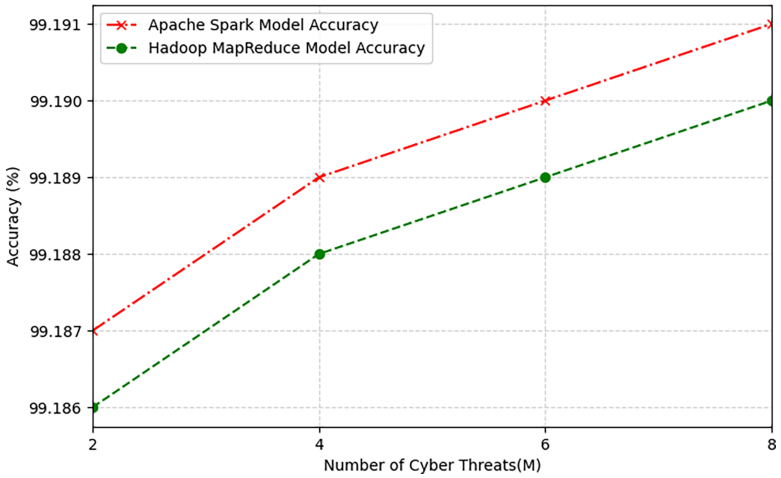

Figure 5: Accuracy comparison of Apache Spark and Hadoop.

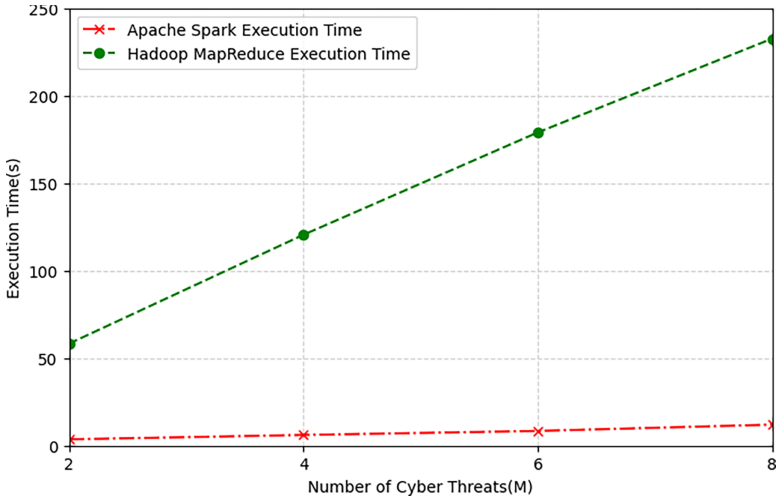

Figure 6: Apache spark and Hadoop MapReduce performance analysis.

3.7 Apache Spark Performance Analysis

In this section, we further extensively evaluate the performance of Apache Spark for computational-based metrics, i.e., scalability and speedup. Spark's scalability was analysed regarding various processing nodes and dataset sizes. Fig. 7 presents the execution times (in Seconds) for processing data of varying sizes (2, 4, 6, and 8M) as the number of nodes in the system increases from 1 to 6. From Fig. 7, as the number of nodes increases, execution time decreases consistently across all data sizes, confirming that distributing the workload across additional nodes improves processing efficiency. For example, processing 2M sequences requires 15 s on a single node but drops to 3.5 s with 6 nodes, while processing 8M sequences reduces from 53 to 12 s under the same scaling conditions. These results confirm that Apache Spark achieves

greater scalability gains on larger datasets, demonstrating its suitability for real-world, large-scale cyber threat detection workloads.

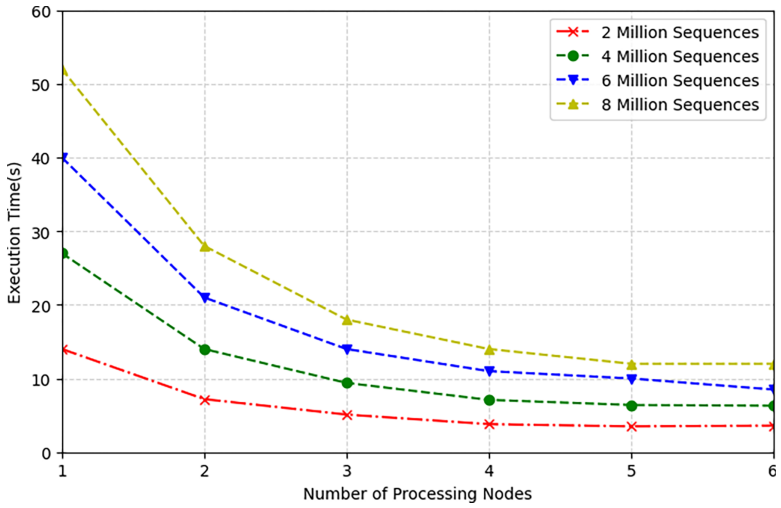

Figure 7: Scalability analysis of Spark using a varied number of processing nodes.

Moreover, different numbers of sequences and processing nodes were used to analyse the speedup of the Spark programming model. The speedup of the Spark can be computed using Eq. (11).

$$S = \frac{T_s}{T_n} \tag{11}$$

S represents the speedup, T_s represents the completion times of the execution on a single Spark node, and T_n represents the completion times on n processing nodes (i.e., n = 6). Based on the results displayed in Fig. 8, the Spark speedup was computed for processing four different numbers of genome samples and varying numbers of processing nodes.

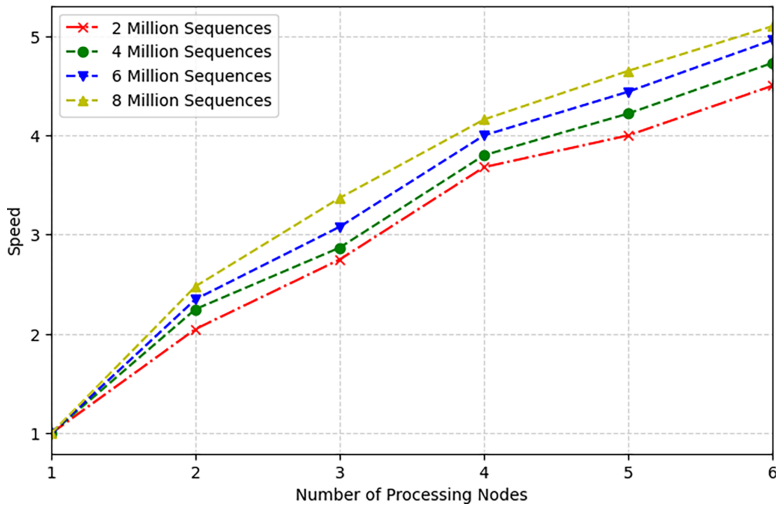

Figure 8: Speedup analysis of Spark on different datasets and processing nodes.

Fig. 8 shows that Spark speedup increases consistently with the number of processing nodes and dataset size. With six nodes, Spark achieves a speedup of 4.50× on 2M sequences and 5.10× on 8M sequences, while on the 6M dataset, speedup grows from 3.08× at three nodes to 4.96× at six nodes. However, Spark does not achieve a linear speedup equal to the number of nodes in any configuration, an expected outcome attributable to communication overhead and task initialisation costs inherent to distributed cluster computing.

4 Conclusion

Social networking sites have a dual nature when it comes to cybersecurity; they can provide intelligence, and they can be used to spread bad information, so the automatic detection and tracking of information relevant to threats is essential for both intelligence development and prompt action on incidents. This paper proposed a scalable framework towards end-to-end deep learning to simultaneously learn a shared semantic representation and a sequential representation of text using Word2Vec and ID-CNN embeddings. The hybrid feature space was refined using SHAP-based feature selection to remove features that are either not informative or redundant, and retain the most predictive features for strong, real-time threat classification using a Deep Neural Network (DNN). Evaluation results using a benchmark dataset showed that the classification performance was quite good, with an accuracy of 99.18% and an MCC of 0.984, better than all the existing baseline methods. For computational scalability, the framework was implemented on two popular Big Data platforms, Apache Hadoop and Apache Spark. The results validated that Apache Spark consistently outperformed Apache Hadoop in all the dataset sizes and node configurations, with a speedup of up to 5.10 times over the largest dataset while sustaining the classification accuracy. Together, these results validate the effectiveness of a hybrid representation learning approach, interpretable feature selection, and scalable big-data processing as a powerful and practical framework for real-time cyber threat detection in large-scale security analytics settings. While the proposed framework demonstrated strong performance on the benchmark dataset, further validation across diverse real-world social media datasets would help confirm its generalizability under varying cyber threat scenarios.

Extended future work will include further optimisation of the framework for even larger data spaces, adaptive hyperparameter tuning, and further optimisation of parallelisation methods to maintain predictive accuracy while further enhancing the throughput efficiency on distributed computing infrastructures.

Acknowledgement: Ongoing Research project (ORF-2026-1181), King Saud University, Riyadh, Saudi Arabia. During the preparation of this manuscript, the authors used AI-assisted writing tools to support grammar checking, sentence restructuring, and language refinement in selected sections of the text. All scientific content, experimental design, data collection, analysis, interpretation of results, and conclusions are entirely the work of the authors. The AI tool was not used to generate ideas, draw conclusions, or contribute intellectually to the research. The authors take full responsibility for the accuracy, integrity, and originality of the work presented in this manuscript.

Funding Statement: Ongoing Research project (ORF-2026-1181), King Saud University, Riyadh, Saudi Arabia.

Author Contributions: All authors contributed equally. Salman Khan and Mai Alzamel wrote the main manuscript, debugged the code, and provided the datasets. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this study are sourced from the publicly available dataset at https://www.kaggle.com/datasets/syedabbasraza/suspicious-tweets/data.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mavaluru D, Mubarakali A, Narapureddy BR, Ramakrishnan J, John R, Ravishankar N, et al. Deep convolutional neural network based real-time abnormal behavior detection in social networks. Comput Electr Eng. 2023;111:108987. doi:10.1016/j.compeleceng.2023.108987.
2. Michael Onyema E, Balasubaramanian S, Suguna SK, Iwendi C, Prasad BVVS, Edeh CD. Remote monitoring system using slow-fast deep convolution neural network model for identifying anti-social activities in surveillance applications. Meas Sens. 2023;27(11):100718. doi:10.1016/j.measen.2023.100718.
3. Dionisio N, Alves F, Ferreira PM, Bessani A. Towards end-to-end cyberthreat detection from twitter using multi-task learning. In: Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN); 2020 Jul 19–24; Glasgow, UK. p. 1–8. doi:10.1109/ijcnn48605.2020.9207159.
4. Santos P, Abreu R, Reis MJCS, Serôdio C, Branco F. A systematic review of cyber threat intelligence: the effectiveness of technologies, strategies, and collaborations in combating modern threats. Sensors. 2025;25(14):4272. doi:10.3390/s25144272.
5. Evangelista JRG, Sassi RJ, Romero M, Napolitano D. Systematic literature review to investigate the application of open source intelligence (OSINT) with artificial intelligence. J Appl Secur Res. 2021;16(3):345–69. doi:10.1080/19361610.2020.1761737.
6. Sun N, Ding M, Jiang J, Xu W, Mo X, Tai Y, et al. Cyber threat intelligence mining for proactive cybersecurity defense: a survey and new perspectives. IEEE Commun Surv Tutor. 2023;25(3):1748–74. doi:10.1109/COMST.2023.3273282.
7. Mahaini MI, Li S. Detecting cyber security related twitter accounts and different sub-groups: a multi-classifier approach. In: Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'21); 2021 Nov 8–11; The Hague, The Netherlands. p. 599–606.
8. Ruohonen J, Hyrynsalmi S, Leppänen V. A mixed methods probe into the direct disclosure of software vulnerabilities. Comput Hum Behav. 2020;103(3):161–73. doi:10.1016/j.chb.2019.09.028.
9. Abdelhaq H, Sengstock C, Gertz M. EvenTweet: online localized event detection from twitter. Proc VLDB Endow. 2013;6(12):1326–9. doi:10.14778/2536274.2536307.
10. Sabottke C, Suciu O, Dumitraş T. Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits. In: Proceedings of the 24th USENIX Security Symposium (USENIX Security'24); 2015 Aug 12–14; Washington, DC, USA. p. 1041–56.
11. Kergl D, Roedler R, Rodosek GD. Detection of zero day exploits using real-time social media streams. In: Advances in nature and biologically inspired computing. Berlin/Heidelberg, Germany: Springer; 2016. p. 405–16. Available from: https://link.springer.com/chapter/10.1007/978-3-319-27400-3_36.
12. Rodriguez A, Okamura K. Generating real time cyber situational awareness information through social media data mining. In: Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC); 2019 Jul 15–19; Milwaukee, WI, USA. p. 502–7. doi:10.1109/compsac.2019.10256.
13. Alves F, Bettini A, Ferreira PM, Bessani A. Processing tweets for cybersecurity threat awareness. Inf Syst. 2021;95(1):101586. doi:10.1016/j.is.2020.101586.
14. Rodriguez A, Okamura K. Enhancing data quality in real-time threat intelligence systems using machine learning. Soc Netw Anal Min. 2020;10(1):91. doi:10.1007/s13278-020-00707-x.
15. Le Sceller Q, Karbab EB, Debbabi M, Iqbal F. SONAR: automatic detection of cyber security events over the twitter stream. In: Proceedings of the 12th International Conference on Availability, Reliability and Security; 2017 Aug 29–Sep 1; Reggio Calabria, Italy. p. 1–11. doi:10.1145/3098954.3098992.
16. Queiroz A, Keegan B, Mtenzi F. Predicting software vulnerability using security discussion in social media. In: Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS 2017); 2017 Jun 29–30; Dublin, Ireland. p. 628–34.
17. Dabiri S, Heaslip K. Developing a Twitter-based traffic event detection model using deep learning architectures. Expert Syst Appl. 2019;118(1):425–39. doi:10.1016/j.eswa.2018.10.017.
18. Alsodi O, Zhou X, Gururajan R, Shrestha A, Btoush E. Cyber threat detection on twitter using deep learning techniques: IDCNN and BiLSTM integration. In: Proceedings of the 2024 Twelfth International Conference on

Advanced Cloud and Big Data (CBD); 2024 Nov 28–Dec 2; Brisbane, Australia. p. 375–9. doi:10.1109/CBD65573.2024.00073.
19. El Amine Bekhouche M, Adi K. Advanced real-time detection of cyber threat information from Tweets. In: Foundations and practice of security. Berlin/Heidelberg, Germany: Springer; 2025. p. 18–33. doi:10.1007/978-3-031-87499-4_2.
20. Dionisio N, Alves F, Ferreira PM, Bessani A. Cyberthreat detection from twitter using deep neural networks. In: Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN); 2019 Jul 14–19; Budapest, Hungary. p. 1–8. doi:10.1109/ijcnn.2019.8852475.
21. Sani AM, Moeini A. Real-time event detection in twitter: a case study. In: Proceedings of the 2020 6th Internationsal Conference on Web Research (ICWR); 2020 Apr 22–23; Tehran, Iran. p. 48–51. doi:10.1109/icwr49608.2020.9122281.
22. Dai F, Hossain MA, Wang Y. State of the art in parallel and distributed systems: emerging trends and challenges. Electronics. 2025;14(4):677. doi:10.3390/electronics14040677.
23. Senger H, Geyer C. Parallel and distributed computing for big data applications. Concurr Comput. 2016;28(8):2412–5. doi:10.1002/cpe.3813.
24. Hueske F, Walther T. Apache flink. In: Encyclopedia of big data technologies. Berlin/Heidelberg, Germany: Springer; 2019. p. 51–8. doi:10.1007/978-3-319-77525-8_303.
25. Szabist, Iqbal MH, Rahim Soomro T. Big data analysis: Apache storm perspective. Int J Comput Trends Technol. 2015;19(1):9–14. doi:10.14445/22312803/ijctt-v19p103.
26. Henning S, Hasselbring W. Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud. J Syst Softw. 2024;208(12):111879. doi:10.1016/j.jss.2023.111879.
27. Khan M, Jin Y, Li M, Xiang Y, Jiang C. Hadoop performance modeling for job estimation and resource provisioning. IEEE Trans Parallel Distrib Syst. 2016;27(2):441–54. doi:10.1109/TPDS.2015.2405552.
28. Tahmassebi A. iDeepLe: deep learning in a flash. Disruptive Technol Inf Sci. 2018;10652:106520S. doi:10.1117/12.2304418.
29. Siva Prasad BVV, Sucharitha G, Venkatesan KGS, Patnala TR, Murari T, Karanam SR. Optimisation of the execution time using hadoop-based parallel machine learning on computing clusters. Lect Notes Data Eng Commun Technol. 2022;117(1):233–44. doi:10.1007/978-981-19-0898-9_18.
30. Khan M, Salman, Iqbal N. Computational performance analysis of cluster-based technologies for big data analytics. In: Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData); 2017 Jun 21–23; Exeter, UK. p. 280–6. doi:10.1109/iThings-GreenCom-CPSCom-SmartData.2017.239.
31. Khan S, Dilshad N, Ahmad N, Al Qahtani SA. Enhancing security information and event management with W2V-BERT-based real-time threat detection. Sci Rep. 2026. doi:10.1038/s41598-026-49610-z.
32. Ouchene L, Bessou S. AlgVec: a word embedding model for the Algerian dialect in Arabic and Arabizi. J King Saud Univ Comput Inf Sci. 2025;38(1):20. doi:10.1007/s44443-025-00407-6.
33. Jahanzeb M, Khan AH, Ahmed S, Alhumam A, Khan MF, Siddiqui SY. Privacy preserving epileptic seizure recognition using federated and explainable machine learning. Discover Comput. 2026;29(1):53. doi:10.1007/s10791-026-09956-4.
34. Khan S, Khan M, Iqbal N, Amiruddin Abd Rahman M, Khalis Abdul Karim M. Deep-piRNA: bi-layered prediction model for PIWI-interacting RNA using discriminative features. Comput Mater Continua. 2022;72(2):2243–58. doi:10.32604/cmc.2022.022901.
35. Khan S, Dilshad N, Ahmad N, Noor S, AlQahtani SA. Integrating AI in security information and event management for real time cyber defense. Sci Rep. 2025;15(1):35872. doi:10.1038/s41598-025-19689-x.
36. Rifano EJ, Fauzan AC, Makhi A, Nadya E, Nasikin Z, Putra FN. Text summarization menggunakan library natural language toolkit (NLTK) berbasis pemrograman python. ILKOMNIKA. 2020;2(1):8–17. doi:10.28926/ilkomnika.v2i1.32.