

Network Attacker Detection Using a Hybrid Graph Neural Network and Temporal Convolutional Network Framework

Titorea Veera Jothi Archunan¹, Pasupathi perumalsamy², Gowsalya Pandiyarajan³

^{1,3}Research scholar

²Professor and Dean

^{1,2,3} Computer Science and Engineering, Sri Vidya College of Engineering & Technology, Virudhunagar, Tamilnadu, India

1. INTRODUCTION

Wireless sensor networks are used for different characteristics than communication networks used for other generic applications. These characteristics are in terms of deployment topology, data processing, environmental conditions and network throughput. The security issues are related to confidentiality, authentication, availability, integrity, authorization and newness. Confidentiality deals with secrecy of data communication. Authentication is necessary for prevention of fake messages from malicious sensor nodes; it ensures data authenticity. Availability means consistency in services in the presence of attacks. Integrity means the data or messages are received in an unaffected form at the destination. Authorization means only authorized sensor nodes can communicate and unauthorized access of data must be prevented. Newness of data is inevitable to ensure that attackers do not replay old data to hinder the security of the network. It is very challenging to ensure data authenticity and security in the network due to the constraints shown in Fig. 1.

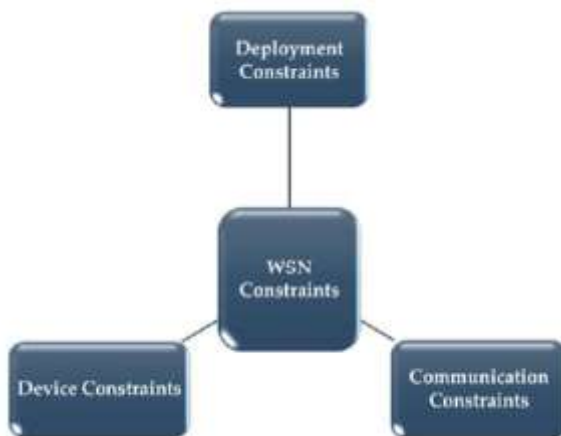


Fig -1: Constraints in wireless sensor networks.

Device Constraints:

Network nodes have very limited storage, processing and computational capabilities. They have limited power as they are battery operated.

Deployment Constraints:

Sensor nodes are deployed in potentially unsecured environments. They are deployed either in a fixed manner or in a random manner, and are remotely managed and controlled.

Communication Constraints:

Networks communicate using radio transmissions and most of them use the unlicensed ISM band, which is also used for many other applications. Co-existence of various wireless standards is a major challenge for secured communication.

Due to the above reasons, sensor nodes are prone to eavesdropping or jamming attacks. Moreover, limited processing and storage capabilities of sensor nodes prevent the use of advanced cryptographic methods, since public key cryptography requires expensive computational techniques. Limited batteries make nodes prone to denial-of-service attacks which can further drain energy. Tampering of sensor nodes and reprogramming of chips are possible attacks, and security can be endangered by reverse engineering. The network also employs various devices and communication protocols at different levels, so interoperability and security with other devices is a matter of concern. The protection and privacy of the network must be treated carefully for smart-grid applications, and the cost-security trade-off must be carefully implemented.

1.1 Classification of Cyber-Attacks in the Network

A. Attacks Based on Location of an Intruder

External Attack: Most cyber-attacks are external in nature, where an intruder is outside the range of the network. External attacks are performed by jamming the network, exhausting the resources or denial of service (DoS).

Internal Attack: In this type of attack, an intruder is in the range of the network. It is performed by physical tampering of a node, revelation of confidential information, and causing denial of service to authorized nodes.

B. Attack Based on Access Level of an Intruder

Active: An adversary performs the attack through modification and theft of data, hampering the confidentiality, availability and integrity of data by packet modification, false data injection, resource exhaustion and node capturing. Spoofing, jamming, wormhole, hello flood, DoS and sink hole are examples of active attacks.

Passive: Passive attacks are performed mainly through observation of network activities, impeding confidentiality. This includes analysis of traffic, decryption of vulnerable data, and capture of information.

C. Attacks on Various Network Layers

Denial of Service Attack: Restricted memory and less computational capacity make networks vulnerable to denial-of-service attacks, in which network resources are made inaccessible by congestion. Three forms are described in the literature: physical devastation or node capture of wireless sensor nodes; utilization of scarce network resources by intruders to make them unavailable; and alteration of network configurations such that encryption and other settings become unapproachable for genuine users.

Misdirection Attack: Information is routed towards a fake path, altering routing information and affecting communication

adversely. Authentication between transmitter and receiver and multi-hop routing can detect it.

Selective Forwarding: A counterfeit node acts like an actual node, diverts packets to a wrong path and selectively drops some packets so that intrusion is difficult to identify. Acknowledgement-based routing, multi data flow and neighbour-based detection can be used.

Sink Hole Attack: A data-link-layer attack in which an intruder colludes with a sensor node or introduces a fake node that attracts network traffic, then drops or alters packets.

Sybil Attack: A malicious node takes multiple identities to disturb the routing and communication process that normally depends on node cooperation.

Wormhole Attack: A data-link-layer intrusion in which a malicious node records all information and diverts it to a wrong path, even without knowledge of the cryptography of the actual node.

Hello Flood Attack: Routing protocols use Hello packets to detect neighbours. Fake packets camouflage hello packets to attract sensor nodes; attackers with ample radio resources can generate this attack, and the victim node identifies a false hello packet as a normal node.

1.2 Intrusion Detection System

Encryption and authentication are unavoidable for the security of any communication system and are vital for reliability, confidentiality and integrity. Detection of attacks is also imperative to make the system robust. An intrusion detection system (IDS) is a subsequent security measure and defence mechanism against cyber-attacks. It observes the network and identifies anomalous activities, and can be defined as a unification of hardware and software tools meant to detect internal or external cyber-attacks. IDS is essential for fault tolerance, security and reliability, and it also investigates physical tampering of sensor nodes, ensuring secured routing of information. The principal tasks of an IDS are prevention and detection of attacks, situational awareness, evidence collection, and administration of connection topologies.

The main components of an IDS are the Sensor, which collects statistics from the monitored system; the Detector, which analyses collected data to identify intrusions; the Information Base, which supports the detector by providing attack signatures; and the Response Manager, which manages responses to cyber-attacks. The general block diagram of an IDS is shown in Fig. 2.



Fig -2: General block diagram of an intrusion detection system.

Intrusion detection approaches for the network are classified as follows. Anomaly Detection identifies node anomalies (failure of a node or power problems), network anomalies (unexpected fluctuations in signal strength and connectivity), and data anomalies (chaotic or disordered data communication); these may be handled with game-theoretic, statistical, machine-learning, artificial-immune-system and data-mining approaches. Misuse Detection is a signature-based system to discover recognized attacks but cannot detect unknown attacks. Hybrid Detection combines anomaly and misuse approaches for

accurate, application-specific results. In addition to IDS, an Intrusion Detection and Prevention System (IDPS) can be developed for rapid and effective mitigation of attacks.

2. LITERATURE SURVEY

Iwendi et al. (2018) presented an improved key management technique (KMT) with ant colony optimization (ACO) for path planning of network nodes, designing a route that deceives intruders while safeguarding data collection between the base station and the nodes. A non-uniform distribution of KMT with initial pheromone and a direction-based selection strategy improves the path search, and the pheromone disappearance coefficient is segmented to balance convergence speed and communication.

Sun et al. (2018) presented a trust ant colony routing algorithm by introducing a node trust evaluation model based on D-S evidence theory into the ant colony routing protocol. Consistent intensity preprocesses conflicting evidence before the D-S combination rule, and nodes with high trust values are selected as routing nodes; simulations in NS2 evaluate packet loss rate and average end-to-end delay under inside malicious node attacks.

Patil et al. (2018) implemented a secure routing technique using multi data flow topologies (MDT) to defend against attacks, with optimization methods that minimize energy cost while keeping security sufficient. Tasks are scheduled according to sensor node locations and target node trustworthiness, optimizing network lifetime and security against black-hole attacks.

Yin et al. (2018) illustrated a clustering routing protocol to achieve better operation performance of the wireless sensor network. Data fusion technology enhances routing security so that the topology structure of the smart grid and the security of node information are effectively improved.

Alkalbani et al. (2015) presented a hybrid solution combining node trustworthiness with an energy-aware routing mechanism, providing security and minimizing power consumption during routing, with performance measured through simulation and comparison with well-known mechanisms.

Maheswari et al. (2016) focused on the Hello Flood attack and proposed a robust formally analyzed routing protocol for network deployment (RAEED) that uses an improved bidirectional verification scheme to rectify the attack.

Nandhini et al. (2017) presented a metaheuristic hybrid algorithm combining the firefly algorithm with Localizability Aided Localization (LAL) routing for monitoring a wireless area, minimizing time latency, packet drop and packet loss compared with traditional methods.

Tokala et al. (2018) developed a security algorithm to route data when the sink node is mobile, proposing the LISA algorithm with the Steiner Minimal Tree algorithm to overcome sniffer, man-in-the-middle and eavesdropping attacks in military fields while providing confidentiality and integrity.

Patle et al. (2016) surveyed vulnerabilities, attack effects and security schemes, focusing on security issues related to network routing protocols and the difficulty of protecting routing against malicious activities.

Kalnoor et al. (2016) focused on QoS-based multipath routing for intrusion detection of the sinkhole attack, providing reliability, congestion control, energy efficiency and bounded end-to-end delay through secured routing combined with intrusion detection.

Mathurkar et al. (2020) evaluated network performance using the AOMDV routing protocol, studying wormhole attack handling and proposing detection and prevention based on round-trip time and other wormhole features.

Ouafaa et al. (2016) analyzed secure hierarchical routing protocols, simulating them in NS2 and comparing them in terms of security, with security issues and solutions discussed.

Saikia et al. (2017) presented Secure Energy Aware Multipath Routing with Key Management (SEAM-KM), using the base station for shortest-path selection, routing-table generation and pairwise key distribution to prolong node lifetime and increase resilience.

Mathur et al. (2015) addressed problems of wireless routing between nodes of a medical network, focusing on selective forwarding attacks and describing a secure routing protocol, validated by test-bed simulation, capable of detecting and correcting the attack.

3. DOMAIN DESCRIPTION AND EXISTING MODELS

There are two forms of data analysis that can be used for extracting models describing important classes or for predicting future data trends: classification and prediction. Classification models predict categorical class labels, while prediction models predict continuous-valued functions. For example, a classification model can categorize bank loan applications as safe or risky, while a prediction model can predict customer expenditure on computer equipment given income and occupation.

3.1 Classification and Prediction

In classification, a model or classifier is constructed to predict categorical labels such as risky or safe for loan data and yes or no for marketing data. In prediction, a model predicts a continuous-valued or ordered value, for example the amount a customer will spend during a sale. Data classification includes two steps: building the classifier (the learning phase, in which classification algorithms build the classifier from a training set of database tuples and their associated class labels) and using the classifier for classification (in which test data estimates the accuracy of classification rules before they are applied to new tuples).

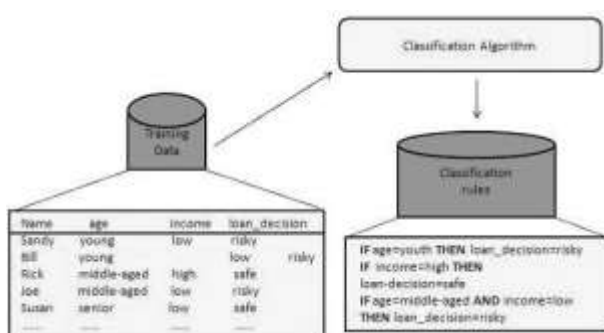


Fig -3: Building the classifier or model.

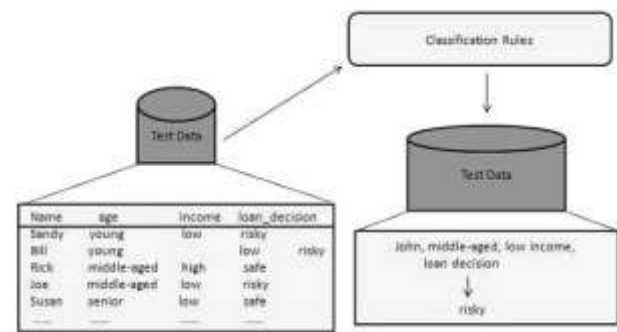


Fig -4: Using the classifier for classification.

3.2 Classification and Prediction Issues

Preparing the data is the major issue and involves: Data Cleaning, removing noise by smoothing and replacing missing values with the most commonly occurring value; Relevance Analysis, using correlation analysis to identify related attributes; and Data Transformation and Reduction, including normalization (scaling values into a small specified range) and generalization (transforming data to higher-level concepts using concept hierarchies). Data can also be reduced by wavelet transformation, binning, histogram analysis and clustering.

Classification and prediction methods are compared by accuracy (correctly predicting class labels or values), speed (computational cost of generating and using the model), robustness (correct predictions from noisy data), scalability (efficient construction given large data) and interpretability (the level of understanding the model provides).

3.3 Types of Algorithms in Data Mining



Fig -5: Data mining algorithms.

Statistical Procedure Based Approach: Classical statistical procedures are characterized by a precise underlying probability model that provides a probability of belonging to each class. Some human involvement is assumed for variable selection, transformation and overall structuring of the problem.

Machine Learning-Based Approach: Automatic computing procedures based on logical or binary operations learn a task from a series of examples. Decision-tree approaches derive classification from a sequence of logical steps and can represent complex problems; genetic algorithms and inductive logic procedures are under active development.

Neural Network: Neural networks consist of layers of interconnected nodes, each producing a non-linear function of its input. They recognize patterns and make simple decisions, and are used in applications such as autopilots and quality control as well as classifying data as intrusive or normal.

Classification Algorithms in Data Mining: A two-step process in which a model is built from training data and then tested against a predefined test set to measure performance and accuracy, assigning class labels to data whose class is unknown.

ID3 Algorithm: Starting with the original set as the root node, on each iteration it computes the entropy of every unused attribute, selects the attribute with the smallest entropy, and splits the set into subsets, recursing on each subset using the remaining attributes.

C4.5 Algorithm: An extension of ID3 that handles both continuous and discrete attributes and missing values. It selects, at each node, the attribute with the highest normalized information gain to split the samples, building a decision tree used as a statistical classifier.

K Nearest Neighbors (KNN): The class of an unknown point is determined from its k nearest neighbours whose classes are known. Variants assign weights according to distance, and structures such as ball trees, k-d trees and orthogonal search trees improve memory use and speed.

Naive Bayes Algorithm: Based on Bayes theorem, it is used when the dimensionality of the inputs is high. It assumes that the presence of a particular feature of a class is unrelated to the presence of other features, computing the posterior probability of a class given the predictors.

SVM Algorithm: Support Vector Machines determine decision boundaries (hyperplanes) that produce the optimal separation of classes with the largest possible margin. Based on statistical learning theory and structural risk minimization, they can use kernel functions to learn non-linear decision functions and are extended to multi-class problems by decomposing them into binary problems.

ANN Algorithm: Artificial neural networks approximate functions of many inputs through interconnected neurons organized into input, hidden and output layers. They are trained by adjusting weights through back-propagation and are well suited to pattern and speech recognition, with parallel architectures offering speed advantages on large continuous data streams.

4. PROPOSED SYSTEM

The proposed system leverages a hybrid deep learning framework combining Graph Neural Networks (GNNs) and Temporal Convolutional Networks (TCNs) to detect cyberattacks in dynamic network environments. It first collects diverse network telemetry data, including traffic flows, system events and firewall logs, covering both normal and malicious activities. The raw data undergoes preprocessing, where relational and temporal features are extracted, missing values handled, and graph structures constructed in which network entities are nodes and communication links are edges. The GNN component learns spatial and relational dependencies among devices, while the TCN component models temporal patterns of network behaviour, enabling detection of sophisticated attacks that evolve over time. Evaluated on benchmark intrusion detection datasets, the system outperforms traditional machine learning algorithms and existing deep learning models in accuracy, detection speed and adaptability.

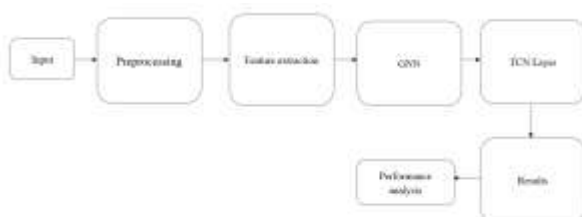


Fig -6: Proposed GNN-TCN system architecture.

The system is composed of several key modules. The Data Collection and Preprocessing Module gathers comprehensive telemetry and produces a cleaned, structured dataset. The Graph Construction Module builds a dynamic graph that captures spatial and relational dependencies. The Feature Extraction and Temporal Encoding Module extracts temporal and relational feature vectors for the TCN. The Graph Neural Network Module learns how devices are connected and interact, producing graph embeddings. The Temporal Convolutional Network Module models long-range dependencies in network behaviour over time, producing temporal embeddings. The Hybrid GNN-TCN Fusion Module fuses graph and temporal embeddings into a joint representation. The Attack Detection and Classification Module classifies activities as normal or malicious with confidence scores. The Evaluation and Performance Assessment Module applies standard metrics and compares the model with existing methods.

4.1 Model Architecture

Input Layer - Syscall Sequence Embedding: Sequences of system calls (syscalls) are encoded as integers and passed through an embedding layer (nn.Embedding) to produce dense vectors that capture semantic relationships between syscalls.

Graph Construction (GNN Input): A graph is built where nodes represent system calls and edges represent the relationships between consecutive syscalls, producing a graph object with node features (x), edge connections (edge_index) and label (y).

Graph Neural Network Layers: SAGEConv layers perform graph-based message passing, aggregating feature information from neighbouring nodes. A first layer learns feature representations from the syscall embeddings and a second layer refines the node information to capture spatial dependencies.

Global Mean Pooling Layer: global_mean_pool aggregates node-level features into a fixed-size, graph-level feature vector summarizing the entire syscall sequence.

Temporal Convolutional Network Layer: A series of 1D dilated convolutions (nn.Conv1d) increases the receptive field and models long-range temporal dependencies, producing a temporal feature map of sequential patterns.

Fully Connected Classifier Layer: A dense layer (nn.Linear) followed by a softmax activation produces a probability distribution over the benign and malicious classes.

Output Layer: The class with the highest probability is the model prediction, either Benign (0) or Malicious (1).

Training and Loss Function: The model is trained to minimize the Cross-Entropy Loss between predicted probabilities and true labels, using the Adam optimizer to adjust the model weights.

5. SOFTWARE SPECIFICATION

5.1 Software and Hardware Requirements

Software required: IDLE 1.7 and Python 1.7.6. Hardware required: a Windows system, a processor up to 1.5 GHz, and up to 512 MB RAM.

5.2 Python

Python began in the late 1980s when Guido van Rossum started a project intended as a successor to the ABC language, with better exception handling and the ability to interface with the Amoeba operating system. It first appeared in 1991, with Python 2.0 released in 2000 and Python 3.0 in 2008. According to the IEEE 2016 ranking, Python ranked third after C and Java, and it is widely used across cloud platforms, mobile app

development, big data, IoT, and block-chain technology. Key features include platform independence, a large standard library, expressiveness, open-source availability, compact code, and dynamic typing. Python is used extensively in platforms such as YouTube, special effects, traffic-control systems, ERP, e-commerce and CRM systems. Major implementations include CPython, Jython, IronPython, MicroPython and PyPy, and it can be installed on Windows, Linux and Mac OS.

6. SIMULATION RESULTS

6.1 Dataset Description

The dataset consists of system call traces from both benign and malicious activities, designed to train and evaluate a model for intrusion detection. Benign traces are system call sequences from normal operations such as routine user interactions and legitimate processes, collected from a variety of environments and labelled as the negative class (0). Malicious traces are system call sequences associated with malware such as trojans, worms and viruses that perform harmful operations, labelled as the positive class (1). Raw traces are stored as text files with one system call per line, and sequence lengths vary with the activity or malware behaviour.

To improve performance and address class imbalance, the benign dataset undergoes data augmentation, generating synthetic traces by introducing small perturbations such as random substitutions or insertions of system calls. Each syscall is converted to an integer encoding using a vocabulary, and the sequences are transformed into a graph structure where each node is a syscall and edges represent sequential dependencies. The data is pre-processed to handle missing values, standardize sequence lengths and filter out sequences that are too short. After preprocessing and augmentation, the dataset contains a balanced distribution of benign and malicious traces, split into training and testing subsets.

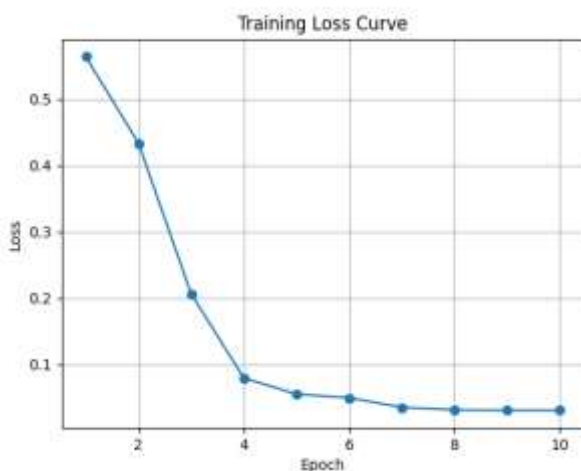


Fig -7: Model training loss curve.

```
>>>
----- RESTART: D:\sa\2023\code\new\LR\testingcatboost.py
Sample 1: No Attack
Sample 2: Attack
Sample 3: No Attack
Sample 4: No Attack
Sample 5: No Attack
Sample 6: Attack
Sample 7: Attack
Sample 8: Attack
Sample 9: Attack
Sample 10: Attack
Sample 11: Attack
Sample 12: Attack
Sample 13: Attack
Sample 14: Attack
Sample 15: Attack
Sample 16: Attack
Sample 17: Attack
Sample 18: Attack
Sample 19: Attack
>>>
```

Fig -8: Model validation output.

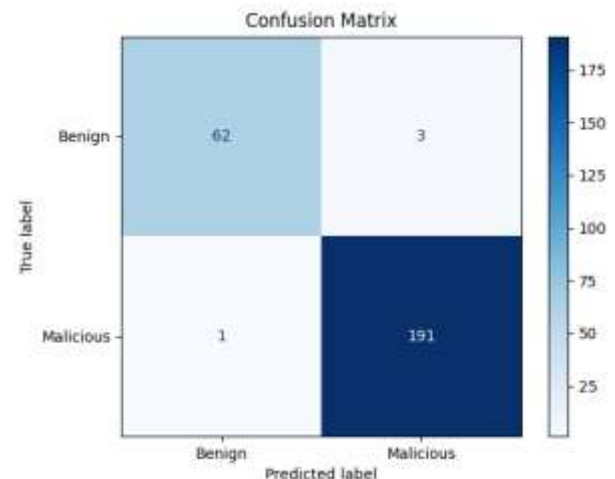


Fig -9: Confusion matrix.

6.2 Performance Analysis

Table -1: Performance comparison of models

Models	F1-Score	Accuracy	Precision	Specificity
NN	89.46	84.43	87.76	78.18
SVM	92.92	89.76	92.13	85.39
Ada boost	94.76	94.84	95.2	92.47
Proposed	96.76	98.84	97.2	95.47

The performance comparison highlights the efficacy of the proposed method. The Neural Network achieved an F1-score of 89.46%, accuracy of 84.43% and precision of 87.76%, but its specificity of 78.18% indicates a relatively higher false-positive rate. The Support Vector Machine improved on this with an F1-score of 92.92%, accuracy of 89.76%, precision of 92.13% and specificity of 85.39%. AdaBoost further enhanced performance with an F1-score of 94.76%, accuracy of 94.84%, precision of 95.2% and specificity of 92.47%.

The proposed model surpasses all the others with an F1-score of 96.76%, an accuracy of 98.84%, a precision of 97.2% and a specificity of 95.47%, indicating a superior ability to differentiate between normal and attack traffic with minimal false positives. This makes the proposed model exceptionally robust in scenarios requiring high precision and reliability in real-time detection, providing a more accurate and reliable solution against both known and emerging network threats.

7. CONCLUSIONS

In this work, we proposed a novel hybrid deep learning approach combining Graph Neural Networks (GNNs) and Temporal Convolutional Networks (TCNs) for effective detection of malicious activities in network environments based on system call sequences. By leveraging the spatial dependencies captured by GNNs and the temporal patterns modelled by TCNs, the approach robustly detects sophisticated and evolving cyberattacks that may not be easily identified using traditional methods. The dataset of benign and malicious system call traces provided a comprehensive basis for training and evaluation, and the data augmentation applied to the benign dataset improved generalization and addressed class imbalance.

Experimental results show that the GNN-TCN hybrid model significantly outperforms conventional machine learning algorithms and other state-of-the-art deep learning models in accuracy, detection speed and adaptability to novel attack

patterns. The ability to handle both spatial and temporal features positions the proposed system as a promising solution for next-generation network intrusion detection systems. In future work, we aim to expand the framework to more complex attack scenarios and larger datasets, explore real-time intrusion detection in operational environments, and integrate more advanced graph-based and temporal analysis techniques to further enhance robustness and efficiency.

ACKNOWLEDGEMENT

The authors would like to thank their institution and the faculty of the Department for their continuous support, guidance and provision of resources during the course of this research work.

REFERENCES

- [1] R. Janapati, C. Balaswamy and K. Soundararajan, "Enhancement of localized routing using CDPSO in WSN," 2018 Conference on Signal Processing and Communication Engineering Systems (SPACES), Vijayawada, 2018, pp. 16-19.
- [2] I. Daanoun, A. Baghdad and A. Ballloul, "A comparative study between ACO-based protocols and PSO-based protocols in WSN," 2019 7th Mediterranean Congress of Telecommunications (CMT), Fès, Morocco, 2019, pp. 1-4.
- [3] M. Shinde and D. C. Mehetre, "Black Hole and Selective Forwarding Attack Detection and Prevention in WSN," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBE), Pune, 2017, pp. 1-6.
- [4] A. Karthikeyan, V. P. Arunachalam, S. Karthik and P. Dhivya, "Energy Efficient Structure Free and Location Based Routing Protocol in WSN," 2018 International Conference on Soft-computing and Network Security (ICSNS), Coimbatore, 2018, pp. 1-5.
- [5] A. Mathur and T. Newe, "Medical WSN: Power, routing and selective forwarding defense," 2015 13th International Conference on Telecommunications (ConTEL), Graz, 2015, pp. 1-6.
- [6] M. Saikia, U. K. Das and M. A. Hussain, "Secure energy aware multi-path routing with key management in wireless sensor network," 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, 2017, pp. 310-315.
- [7] I. Ouafaa, E. Mustapha, K. Salah-ddine, L. Jalal and E. H. Said, "An advanced analysis on secure hierarchical routing protocols in wireless sensor network," 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, 2016, pp. 1-6.
- [8] A. V. Mathurkar and P. D. Dorge, "Performance Evaluation of Network Using AOMDV Routing Protocol," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO), Noida, India, 2020, pp. 479-483.
- [9] G. Kalnoor and J. Agarkhed, "QoS based multipath routing for intrusion detection of sinkhole attack in wireless sensor networks," 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, 2016, pp. 1-6.
- [10] A. Patle and N. Gupta, "Vulnerabilities, attack effect and different security scheme in WSN: A survey," 2016 International Conference on ICT in Business Industry & Government (ICTBIG), Indore, 2016, pp. 1-6.
- [11] M. Tokala and R. Nallamekala, "Secured algorithm for routing the military field data using Dynamic Sink: WSN," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018, pp. 471-476.
- [12] M. Nandhini and P. Priya, "A hybrid routing algorithm for secure environmental monitoring system in WSN," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2017, pp. 1061-1065.
- [13] S. U. Maheswari, N. S. Usha, E. A. M. Anita and K. R. Devi, "A novel robust routing protocol RAEED to avoid DoS attacks in WSN," 2016 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, 2016, pp. 1-5.
- [14] A. S. Alkalbani, T. Mantoro, S. Moedjiono and M. A. Ayu, "Hybrid Solution for Secured and Energy Aware Routing Path in Wireless Sensor Networks," 2015 Fourth International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec), Jakarta, 2015, pp. 78-83.
- [15] H. Yin, Z. Yin, Y. Yang and J. Sun, "Research on the Node Information Security of WSN Based on Multi-Party Data Fusion Algorithm," 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, 2018, pp. 400-405.
- [16] B. Patil and R. Kadam, "A novel approach to secure routing protocols in WSN," 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2018, pp. 1094-1097.
- [17] Z. Sun, Z. Zhang, C. Xiao and G. Qu, "D-S evidence theory based trust ant colony routing in WSN," China Communications, vol. 15, no. 3, pp. 27-41, March 2018.
- [18] C. Iwendi, Z. Zhang and X. Du, "ACO based key management routing mechanism for WSN security and data collection," 2018 IEEE International Conference on Industrial Technology (ICIT), Lyon, 2018, pp. 1935-1939.
- [19] S. Karthick, E. S. Devi and R. V. Nagarajan, "Trust-distrust protocol for the secure routing in wireless sensor networks," 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), Chennai, 2017, pp. 1-5.
- [20] I. A. Umar, Z. M. Hanapi, A. Sali and Z. A. Zulkarnain, "A Forwarding Strategy for DWSIGF Routing Protocol," 2015 5th International Conference on IT Convergence and Security (ICITCS), Kuala Lumpur, 2015, pp. 1-4.
- [21] A. Karakaya and S. Akleylek, "A survey on security threats and authentication approaches in wireless sensor networks," 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, 2018, pp. 1-4.
- [22] L. Yi and F. Zhongyong, "The Research of Security Threat and Corresponding Defense Strategy for WSN," 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation, Nanchang, 2015, pp. 1274-1277.