International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Smart Intrusion Detection Scheme for Data Transmission Over WSN

Pooja Yadav, Dr. C.L.P. Gupta
Department of Computer Science and Engineering,
Bansal Institute of Engineering & Technology, Lucknow – India
poojay1799@gmail.com, clpgupta@gmail.com

Abstract: Anomaly (outlier) detection is critical in ESN-based monitoring applications that utilise big data sets for biomedical and defence applications. Temperature, humidity, sound, pressure, vibration, and other environmental and physical factors are monitored via a wireless sensor network.
Due to limitations in energy, memory, computing power, and bandwidth, along with the dynamic nature of the network and harsh environmental conditions, large collections of sensor nodes in a WSN often produce raw sensor data with low quality and reliability. This paper presents a KNN-based prediction model for outlier detection, utilizing the entropy values of incoming sensor voltages. The model is developed and analyzed using a real-time database generated from 14 sets of MICA2 wireless sensor kits, with anomalies manually introduced in real time by Intel Berkeley Lab volunteers through motion-based intrusions. To create a comprehensive training dataset for outlier detection, a fixed-window segmentation approach is applied to each sensor data pair. The analysis shows that the proposed method achieves an 86% accuracy in identifying outliers. Additionally, the study evaluates the impact of varying distance metrics and the number of nearest neighbors on the KNN model's performance.

Keywords— WSN, Outliers Detection, Anomaly Detection, KNN

1. Introduction:

Outlier detection plays a crucial role in data analysis by identifying data points that substantially deviate from the expected patterns or distributions. These anomalies, often referred to as outliers or exceptions, may emerge due to several factors, including sensor malfunctions, measurement inaccuracies, data entry errors, or even rare but significant events. Detecting and addressing such deviations is essential for ensuring data integrity, improving predictive accuracy, and enhancing decision-making processes across various domains. Detecting such anomalies is essential for ensuring data integrity and improving the performance of models, especially in fields like fraud detection, network security, and medical diagnosis.
Outlier detection techniques are designed to distinguish these abnormal points from the majority of the data, allowing for cleaner datasets, more accurate predictions, and more reliable conclusions.
The selection of an outlier detection technique largely depends on the characteristics of the dataset, including its structure, dimensionality, and the specific application requirements. Different methodologies are employed based on these factors to ensure effective anomaly identification. Generally, outlier detection approaches can be classified into four main categories: statistical methods, which rely on probability distributions; proximity-based techniques, which assess the distance between data points; clustering-based approaches, which identify deviations from established groups; and machine learning-based models, which leverage pattern recognition and predictive algorithms to detect anomalies. Statistical methods rely on predefined distribution models, while proximity-based techniques examine the distance between points to detect anomalies. Clustering methods classify outliers as points that do not fit into any cluster, and machine learning approaches leverage models to automatically detect deviations without requiring predefined thresholds.
Each technique offers distinct advantages and limitations depending on the data's complexity and the computational resources available. As datasets continue to grow in size and complexity, the need for advanced, efficient, and scalable outlier detection techniques becomes increasingly critical for modern data-driven applications.

Types of Outlier Detection Techniques

Outlier detection methods can be broadly classified according to their fundamental principles, the assumptions they make about the data, and the types of input data they are designed to process. These techniques are tailored to different contexts, allowing for more accurate identification of anomalies based on the specific characteristics of the dataset. The primary categories of outlier detection methods include:

1. **Statistical-Based Techniques**: These techniques are based on probabilistic models that help identify data points that significantly deviate from the expected distribution. They are particularly useful in scenarios where the data is assumed to follow a specific statistical distribution, such as a normal distribution. By comparing the observed data points to the expected behavior defined by the model, these methods can effectively highlight outliers that do not conform to the established pattern, these methods employ measures such as Z-scores, Grubbs' test, and boxplots to detect outliers. While easy to implement for univariate data, statistical methods can become inefficient for high-dimensional or non-Gaussian data, as they assume predefined distribution properties that may not hold in real-world datasets.

2. **Proximity-Based Techniques**: These methods, such as k-nearest neighbors (KNN) and distance-based algorithms, identify outliers based on the remoteness between data points. The assumption here is that outliers are far from their nearest neighbors in the feature space. These strategies work well for low-dimensional data but can suffer performance challenges in high-dimensional areas due to the "curse of dimensionality," where the concept of distance becomes less relevant as the number of dimensions grows.
3. **Clustering-Based Techniques**: In clustering-based methods, outliers are typically recognized as data points that either do not have its place to any cluster or are part of small, sparse clusters that are significantly less dense than the rest of the data. These techniques leverage the structure of the data to separate points into meaningful groups, with outliers being those that fall outside these clusters. Common algorithms employed for outlier detection in this context include DBSCAN, which identifies ranges of high density and marks points in low-density regions as outliers, and k-means clustering, where outliers are often those points that are far from the centroids of the formed clusters. These techniques work well when the data naturally clusters into groups, but they are sensitive to the choice of parameters like the number of clusters, which can affect their accuracy in identifying anomalies.
4. **Techniques based on ML**: As the complexity of datasets has increased, machine learning has emerged as a powerful tool for outlier detection. Unsupervised, semi-supervised, and supervised approaches are all possible. Supervised techniques require labelled data, where outliers are predefined, while unsupervised methods like Isolation Forest, One-Class SVM, and autoencoders are capable of identifying outliers without prior labelling. Machine learning techniques are highly effective at detecting complex patterns in data but can be computationally affluent and may require big data to train accurately.

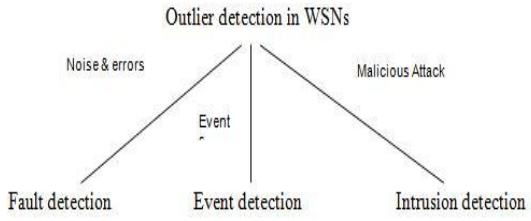

Fig. 1: Sources of outliers in WSN

2. Related Work:

For sensor networks that depend on organizational strategies, such as tracking progress and verifying failures, [4] introduced Intrusion Detection Systems (IDS). The system looks for malicious intrusions by comparing event data with known records. [5] implemented this detection framework in a cluster-based sensor network, similar to the architecture discussed in this paper. The system can identify previously observed anomalies. However, this study focuses on detecting anomalies in more complex scenarios where the system cannot access rare patterns.

To identify unusual traffic patterns, [6] proposed disruption detection schemes that model typical traffic behavior. Their approach can effectively differentiate between attacks and previously observed patterns.

[8] introduced a calculation for identifying anomalies in the context of Bayesian belief networks (BBN). Additionally, the framework is prepared to evaluate the sensor data's missing attributes. Both the spatial transient correlations among the sensor hubs and the relationship between their attributes can be detected by the BBNs.

A technique for detecting anomalies based on k-closest neighbors was developed, where points whose remoteness to their k-NNs exceeds a beginning or whose top n focuses are abnormal are flagged. Each sensor maintains a histogram-like overview of relevant data within a descending window of its most informative components. The sink hub collects these synopses and requests additional data from the network to accurately identify anomalies across the entire system. Unlike synopses, a simple, focused approach facilitates more communication [9]. Several distinctions set their approach apart from ours: they identify anomalies based on a single layer of information, and the complexity of creating smaller histograms limits further improvement. In contrast, our method builds upon these elements, considering additional factors, while theirs focuses only on two k-NN-based anomaly definitions. Furthermore, their approach is limited to minimal spatial proximity, whereas our method can incorporate physical proximity when needed, enabling “semi-nearby” anomaly detection.

Another method needs sensors to maintain a tree-based topology and employs a measure of the underlying probability distribution from whence the data originates to find abnormalities [10]. Each sensor samples its data to compute this measure.

[11] proposed a system based on a BBN that is distributed across the Internet of Things, enabling each sensor to determine the likelihood of a tuple being observed and detect anomalies.

[12] presented a wavelet-based method for rectifying isolated, sizable spikes in input streams from a single sensor. In order to find more consistent temporal patterns in the data, they also compared data streams from geographically nearby sensors using a time-warping distance-based technique.

[14] broke down the many oddity finding processes for remote sensor organizations and focused on the uniqueness of IOT, as well as the useful characteristics of inconsistency detection methodologies.

[15] illustrated the many IOT types and possible solutions for managing the challenges that have been documented and organizing a wide variety of concerns. To help people learn

more about this emerging subject, this article will provide information about IOT and sorts of audit writing.

3. Methodology:

The K-NN approach is used for both classification and regression tasks. It was first created in 1951 by Evelyn Fix and Joseph Hodges, and it was later expanded in the statistical field by Thomas Cover. In this method, the K nearest data points in the training set serve as the basis for determining the result. Depending on the application, K-NN can be used for organization or regression:

In K-Nearest N classification, the item is assigned to the class with the majority of neighbors. The class is determined by a majority vote among the k nearest neighbors (with K typically being a small positive integer).

A property value, usually determined as the weighted sum of the K nearest neighbors, is the result of K-NN regression. The item belongs to the single nearest neighbor's class if K = 1.

The function is locally approximated by K-NN classification, and since it relies on distance metrics, it is important to normalize the training data if the features vary in scale or units, as this can improve performance. A common enhancement is to apply weights to the neighbors, with closer neighbors contributing more to the result than farther ones. Assigning a weight of 1/d, where d is the distance between data points, is a common weighting strategy [8].

The algorithm selects neighbors from a set of items with known classes (in classification) or known values (in regression). While explicit training is not required, this set of items can be considered the training data for the algorithm. One key feature of K-NN is its sensitivity to the spatial relationships within the data.

Algorithm:

The feature vectors and membership functions of the training samples are stored during the training phase, and the training dataset is made up of class-labeled vectors in a subspace. To classify an unlabeled vector (a query or test point), the label of the KNN, based on Euclidean distance, is assigned to the test point. K is a user-defined parameter during classification. For time series, the Euclidean distance is commonly used, while for categorical data like text, other metrics such as the Hamming distance (overlap metric) may be applied. In gene expression microarray data, K-NN can also use correlation measures like Pearson and Spearman [15].

Specialized distance measures, such as Local Components Analysis, can improve the K-NN classification performance. A challenge with the basic "majority voting" approach is class imbalance, where more frequent classes tend to dominate the results [14]. One solution is to weight the classification by considering the distance between the test point and each of its K closest neighbors. In this case, the class (or value in regression) is calculated based on a weighted sum, where closer points contribute more.

Another way to handle class imbalance is by using self-organizing maps (SOMs), where each node represents the center of a cluster of similar data points. K-NN can then be applied to SOMs. The choice of the optimal K value depends on the data; generally, higher values of K reduce the impact of noise but can obscure class boundaries [12]. Heuristic methods, such as hyperparameter optimization, are often used to select the best K. When K = 1, the nearest neighbor method predicts the class based on the closest base classifiers.

The accuracy of K-NN can be significantly affected by noisy or irrelevant features or by feature scales that are disproportionate to their significance. Various studies have focused on selecting or scaling attributes to enhance classification, with evolutionary algorithms being a popular method for feature scaling. Another technique involves using mutual information between training data and class labels to scale features [10]. For binary text classification, it is preferable to use an odd value for K to avoid tied votes, and the bootstrap technique is often used to empirically determine the optimal K [11].

4. Result and Discussion:

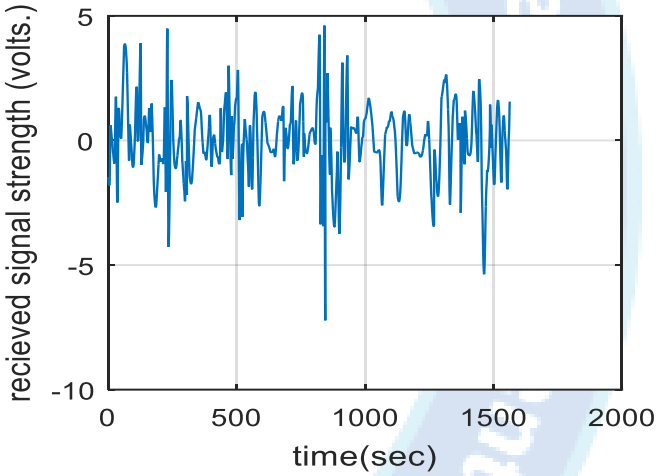

Figure 2: Received signal strength vs time

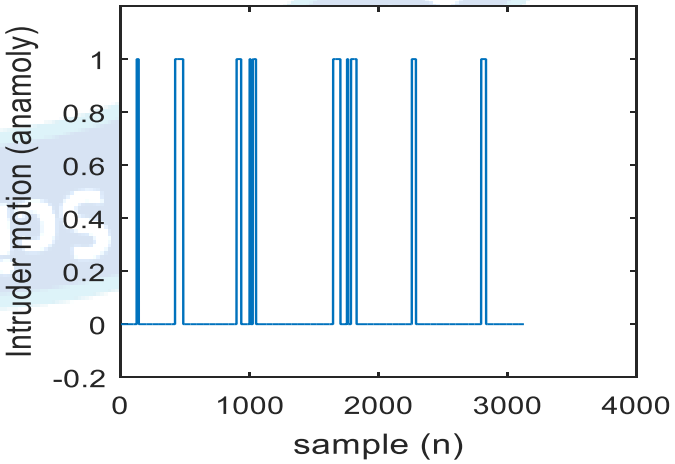

Figure 3: Intruder motion (anamoly) vs sample

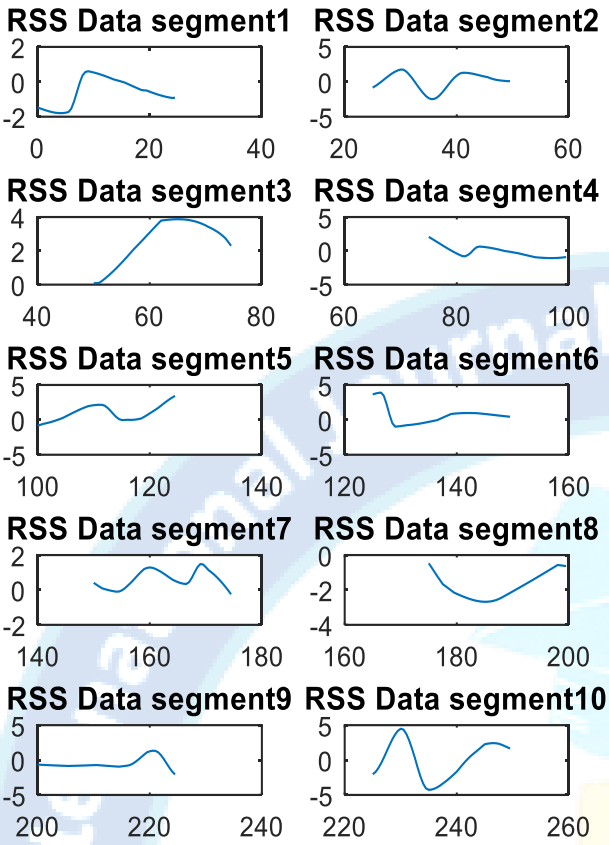

Figure 4: RSS data segments of received data at length of 50 samples.

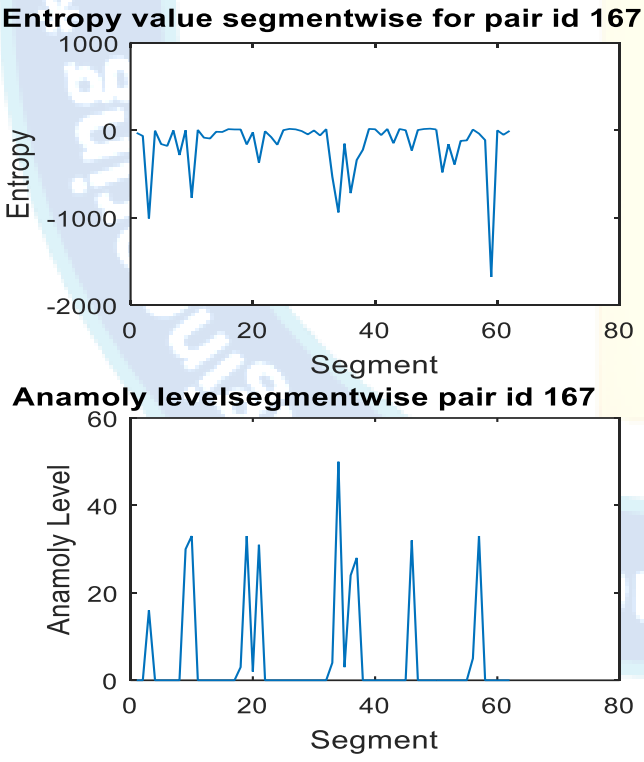

Figure 5: a) Entropy value vs segment of pairid 167, b)Anamoly level segmentwise vs segment of pairid 167.

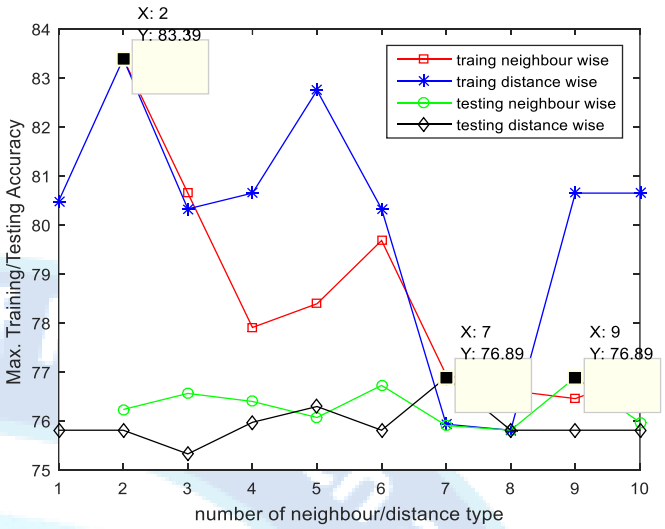

Figure 6: Maximum accuracy

The plot between the highest training and testing accuracy attained during the first attempt at KNN-based detection is displayed in Figure 6. The training accuracy result at various neighbor counts (x axis) is displayed by the red line. There is a range of two to ten neighbors. It has been noted that the two nearest neighbors have the highest training accuracy, with a value of 83.39. The training accuracy at each distance (D1 to D10) is shown by the blue line. The training accuracy in this instance is for the two nearest neighbors at the D2.
Lines that are black for distance type and green for nearest neighbor number indicate the testing accuracy as a percentage. According to the neighbor-wise plot, the distance type D7 for nine neighbors has the maximum testing accuracy, at 76.89%.

5. Conclusion:

In this study, the KNN technique for outlier identification was presented. It explains how to use sensor data records that have been impacted by intrusion-related outliers. It describes the procedures involved in gathering and segmenting data. To obtain the segment-wise entropy values, the data undergoes feature extraction following pre-processing. Lastly, it explains how to use the entropy feature to construct model K closest neighbor for outlier identification.

References:

[1] Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. ACM Comput. Surv. 41, 15, 2009.
[2] Raja Jurdak, X. Rosalind Wang, Oliver Obst, and Philip Valencia,"Wireless Sensor Network Anomalies: Diagnosis and Detection Strategies", 2011.
[3] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Outlier Detection Techniques for Wireless Sensor Network" A Survey,Technical Report, University of Twente, 2008.
[4] Techateerawat, P. and Jennings, "A Energy efficiency of intrusion detection systems in wireless sensor networks", In Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology (WI- IATW), pages 227–230, Washington, DC, USA. IEEE Computer Society, 2006.
[5] Hai, T. H., Khan, F. and nam Huh, E. "Hybrid intrusion detection system for wireless sensor networks",

Computational Science and Its Applications ICCSA, 4706:383–396, 2007.
[6] Onat and Miri, Onat, I. and Miri, "An intrusion detection system for wireless sensor networks", In IEEE International Conference on Wireless And Mobile Computing, Networking and Communications, Los Alamitos, IEEE Computer Society Press, 2005.
[7] Banerjee, Banerjee S., Grosan C., and Abraham, "Ideas: intrusion detection based On emotional ants for sensors", In The 5th International Conference on Intelligent Systems Design and Applications (ISDA), pages 344–349, Wroclaw, Poland, 2005.
[8] Janakiram, D., Reddy, V., and Kumar A., "Outlier detection in wireless sensor Networks using Bayesian belief networks". In First International, Conference on Communication System Software and Middleware, pages 1–6, 2006.
[9]. Patcha, A.; Park, J.-M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Comput. Netw. 2007, 51, 3448–3470.
[10]. Snyder, D. Online Intrusion Detection Using Sequences of System Calls. Master's Thesis, Department of Computer Science, Florida State University, Tallahassee, FL, USA, 2001.
[11]. Markou, M.; Singh, S. Novelty detection: A review—Part 1: Statistical approaches. Signal Process. 2003, 83, 2481–2497.
[12]. Markou, M.; Singh, S. Novelty detection: A review—Part 2: Neural network based approaches. Signal Process. 2003, 83, 2499–2521.
[13]. Hodge, V.; Austin, J. A Survey of Outlier Detection Methodologies. Artif. Intell. Rev. 2004, 22, 85–126.
[14]. Goldstein, M.; Uchida, S. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. PLoS ONE 2016, 11, e0152173.
[15]. Wang, H.; Bah,M.J.; Hammad,M. Progress in Outlier Detection Techniques: A. Survey. IEEE Access 2019, 7, 107964–108000.
[16] Anurag et. al., "Load Forecasting by using ANFIS", International Journal of Research and Development in Applied Science and Engineering, Volume 20, Issue 1, 2020
[17] Raghawend, Anurag, "Detect Skin Defects by Modern Image Segmentation Approach, Volume 20, Issue 1, 2020
[18]. Pimentel, M.A.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A review of novelty detection. Signal Process. 2014, 99, 215–249.
[19]. Chauhan, P.; Shukla, M. A review on outlier detection techniques on data stream by using different approaches of K-Means algorithm. In Proceedings of the 2015 International Conference on Advances in Computer Engineering and Applications, Ghaziabad, India, 19–20 March 2015.
[20]. Salehi, M.; Rashidi, L. A Survey on Anomaly detection in Evolving Data. ACM Sigkdd Explor. Newsl. 2018, 20, 13–23.