Journal of Namibian Studies, 32 (2022): 186-206 ISSN: 2197-5523 (online)

Special Issue 0n Multidisciplinary Research

Machine Learning Approaches To Foster Trust Among Social Iot Devices

Harmanpreet Kaur and Dr. Sonia Vatta

Rayat Bahra University, Mohali.

Abstract

In today's communication era, physical objects or items have gained the ability to communicate with each other and exchange information through the Internet. The advancement has made these objects smarter and capable of interacting with authenticated devices, leading to the emergence of the term "Internet of Things" (IoT). Social IoT (SIoT) devices take this concept further by incorporating social networking paradigms into interactions among smart devices. Achieving socialization among these devices necessitates a secure and trusted connection between them. Consequently, privacy and security enhancements for SIoT devices have become a significant concern and are now essential in contemporary communication.The primary responsibilities within trust management encompass the design of trust architecture and the assessment of reputation. However, applying existing trust architectures and reputation evaluation methods directly to the Internet of Things (IoT) poses challenges. This is primarily because of the huge volume of physical entities involved, the constrained computational capabilities of these entities, and the highly dynamic nature of the IoT network. In this work, we have presented trust management architecture and its evaluation in IoT. It also presents, how trust management helps in reliable decision making process and we have also discussed about various ML approaches used in building trust management system.

Keywords: Machine learning, Artificial Intelligence, Decision Tree, Internet of Things, Social Internet of Things, Support vector machine.

1. Introduction

There are scientific evidences that a large number of individuals tied in a social network can provide far more accurate responses to complex problems than a single individual or a small group of individual. The exploration of this principle has undergone extensive examination in the realm of internet-related research. As a result, many strategies have been put forth which use social networks for purposes such as internet resource search, traffic routing, and the formulation of efficient content distribution policies [1]. IoT combines numerous technologies and predicts large number of objects in our surroundings which are equipped with distinctive addressing mechanisms and standardized communication protocols, possess the capability to engage in interactions and collaborate with neighboring devices to achieve objectives [2]. Life has become smarter, and the Earth is experiencing an improved quality of life due to the widespread of IoT which offers seamless access for anyone, anytime, and anywhere. This technological advancement has ushered in a new era of innovation across various domains, including smarter cities, intelligent transportation, efficient logistics, home automation), precision agriculture, enhanced healthcare, and much more. These revolutionary sectors have reaped the benefits of IoT integration, resulting in increased efficiency and convenience. Furthermore, the IoT has catalyzed a social networking revolution, fostering extensive collaboration among individuals and communities. This interconnectedness has given rise to a wealth of opportunities for people to connect, share, and learn from one another, enriching our global society.

The implemention of social networking principles within the context of the Internet of Things offers several valuable advantages which are given below [3]:

1. Flexible SIoT Structure: The structure of the Social Internet of Things can be adjusted as needed. This customization ensures seamless network navigability, enabling the effective discovery of objects and services. Furthermore, it guarantees scalability, similar to what is observed in human social networks.
2. Reusability: Models originally designed for the analysis of social networks can be repurposed to address IoT-related challenges. These models prove to be particularly useful when dealing with extensive networks of interconnected objects, as they offer valuable insights and solutions.
3. Enhanced Trustworthiness: The incorporation of social networking principles allows for the establishment of trust levels within the IoT. This trust framework can be used to enhance interactions among IoT devices promoting cooperation and reliability within the network.

In other words, we can say that due to merging of the "Internet of Things" and "Social Networks" there is growth in recognition of Social Internet of Things paradigm which is bringing numerous advantages in a future where intelligent objects seamlessly integrate into the daily lives of individuals.The Internet of Things (IoT) is rapidly expanding at a pace unmatched by any other technology.It holds the promise of a world where devices of all kinds, from everyday household appliances to complex enterprise infrastructure, are interconnected. Many companies are investing substantial in the development of the next generation of IoT smart devices. Moreover, researchers are predicting something even more profound: in which all your devices not only track your social connections but also compile your content in a manner that allows for seamless linking and management. This concept is referred to as Social IoT, representing an ecosystem for our interconnected lives. SIoT and its interfaces grant us access to every facet of our interconnected life.

2. Related Work

There has been minimal research conducted in the realm of trust management schemes for SIoT.Trust, being a subjective concept, requires dedicated attention [4]. This section provides an overview of the latest trust management solutions and strategies that have been implemented in SIoT networks.

Chen et al. introduced a trust management framework for IoT systems based on service-oriented architecture (SOA) [5]. This framework relies on past interactions and experiences of entities, utilizing distributed collaborative filtering to make trust recommendations. Furthermore, it dynamically adapts its protocol parameters to suit various environmental conditions while also accounting for four distinct types of malicious attacks.In the same year, Jayasinghe et al. proposed an innovative trust computational model centered around three trust metrics (TMs) for the Social Internet of Things: Knowledge, Recommendations, and Reputations. However, it's worth noting that both the framework by Chen et al. (2016) and the model by Jayasinghe et al. [6] were primarily designed with specific applications in mind, such as service provisioning.

Bernabe et al., primarily focuson trust control techniques [7]. On the other hand, Hellaoui, in his work has introduced an efficient adaptive security model tailored for the Internet of Things (IoT)[8]. This model enables the assessment of trust in relation to the presence of security threats and subsequently facilitates adaptive security decision-making. It's important to note that both the models proposed by Hellaoui et al. (2016) and Bernabe et al. rely on specific assumptions, such as the availability of feedback and knowledge of ownership, among others.

Kotis et al. introduced the notion of "trust semantics," a potent modeling approach designed to streamline the deployment of IoT entities once a suitable selection process has been thoroughly elucidated [9]. Their approach employed fuzzy ontologies and demonstrated the seamless integration of these semantics with IoT ontologies, serving as a secure cornerstone for IoT

applications among the existing entities. Furthermore, their research highlighted the importance of contextual trust in applications and deployments.

Bouazza et al. introduced a novel hybrid approach designed to provide personalized IoT service recommendations to customers. This approach seamlessly blends implicit content-based filtering with ontological elements [10]. In their work, they represented the Social Internet of Things (S-IoT) using ontologies. Collaborative filtering is utilized to predict ratings and generate recommendations; the authors also incorporate social relationships between items into the recommendation process.The evaluation results clearly indicate that the recommended technique excels in terms of personalization and recommendation precision compared to conventional collaborative filtering, especially when leveraged within the context of S-IoT.

Gheisari et al. introduced a three-module architecture based on ontology with the aim of addressing challenges related to trust and heterogeneity. This architecture is designed to handle these challenges while also preserving private information [11]. Data Storage Module is responsible for storing data. It serves as the repository for the information required for subsequent processing.Semantic Analysis and Quality Assurance Module focuses on incorporating semantics into the system to detect unusual or aberrant patterns. Additionally, it ensures that service quality is maintained throughout the process.Privacy and Security Management Module is dedicated to manage security concerns and privacy issues associated with IoT (Internet of Things) devices. It achieves this by dynamically adjusting the privacy settings of these devices as needed to enhance privacy protection.

3. Trust Management Architecture and Evaluation in IoT

Trust management architecture is a critical component within the realm of the IoT as it plays a pivotal role in establishing and sustaining trust among the large

number of IoT components, devices, and entities [12]. The assessment of trust within IoT systems is important for ensuring the reliable and secure operation of these systems. The trust management architecture and its evaluation process in IoT is illustrated in this section:

Trust management architecture in IoT typically involves the following components as shown in Figure 1 [13]:

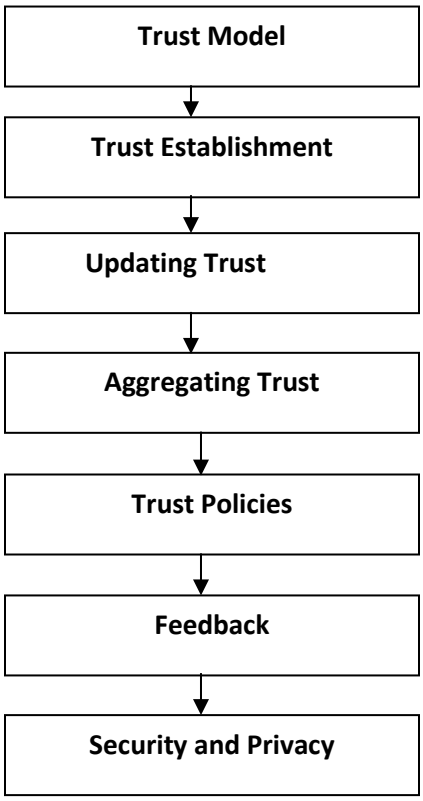

Figure. 1. Architecture of Trust Management

1. Trust Model: It creates a trust framework that outlines how trust is measured, depicted, and overseen within the IoT environment. This framework can encompass factors such as dependability, security, confidentiality, and efficiency.
2. Establishing Trust: Formulates strategies for initiating trust, encompassing processes such as initial device validation, building a reputation, and

verifying identities. Robust cryptographic protocols and secure communication methods play an important role.

3. Updating Trust: Integrate Systems for the dynamic adjustment of trust ratings as devices and entities evolve. Trust levels can fluctuate in response to historical performance and real-time evaluations.
4. Aggregating Trust: Gathers trust data from diverse devices to increase reliability. This process might entail merging data from various devices and appraising the trustworthiness of clusters.
5. Establish Trust Policies: Develops policies that govern how trust decisions are made. These policies will specify thresholds for trust levels and outline responses in cases where trust is compromised.
6. Trust Feedback: Provides feedback to IoT devices and entities regarding their trust status. This can include alerts, notifications, and access control decisions based on trust levels.
7. Security and Privacy: Ensures that trust management processes are secure and respect the privacy of IoT users. Encryption, access control, and data sycrohinzation are critical components.

Assessing trust in IoT systems involves the comprehensive evaluation of trust management mechanisms for their effectiveness and reliability. The key aspects of IoT trust evaluation involves: Trust Metrics which establish clear metrics and criteria to gauge trust within IoT systems. These metrics may include accuracy, the rate of false positives/negatives, trust convergence, and how trust changes over time. Simulation and Testing is conducted thorough simulations and testing to assess the performance of trust management mechanisms under various conditions. This should encompass scenarios of normal operation as well as potential attack scenarios [14,15]. Real-world Deployment will evaluate the trust management mechanisms in real-world IoT deployments. This real-world data collection helps in assessing the trustworthiness, reliability, and security of these mechanisms in practical settings.Monitoring and

Auditing is implemented continuously for monitoring of trust-related events and conduct periodic audits to identify and rectify trust issues within IoT systems. This proactive approach helps in maintaining trust. User Feedback will gather feedback from IoT users to gain insights of their perceptions of trust and their experiences with trust-related decisions made by the system. This user perspective is invaluable in improving trust management. Adaptation and Learning is evaluating the system's ability to adapt to changing conditions and learn from past trust-related incidents. Learning from previous incidents can enhance the system's decision-making process.Compliance and Regulations will ensure that trust management practices align with relevant regulations and standards, such as the General Data Protection Regulation (GDPR) for privacy or ISO/IEC 27001 for security. Compliance is essential for building trust.Performance Impact will assess the performance impact introduced by trust management mechanisms. Ensure that these mechanisms do not significantly degrade system performance while enhancing trust.By implementing well-defined trust management architecture and rigorously evaluating these key aspects, IoT systems can establish and maintain trust among their components and stakeholders. This, in turn, enhances the overall security and reliability of IoT ecosystems.

4. Trust Management in Decision Making

Trust is a vital concept that assists entities in making decisions when faced with uncertain situations. In the computation of trust, we employ the following trust metrics [16].

1. **Direct Trust:** It is determined by an entity's personal interactions with its peers. Specifically, D_{ij} represents node i assessment of node j trustworthiness following their direct interaction. Node i supplies feedback (f_{ij}) to evaluate the quality of the service provided by node j during transaction tl. Additionally, a transactional factor (tlf_{ij}) is

maintained by each node, scoring transactions between them as 1 if relevant and 0 otherwise [17].

$$tlf_{ii} = \begin{cases} 0, & \text{irrelevant transaction} \\ 1, & \text{relevant transaction} \end{cases}$$

$$f_{ii} = \begin{cases} 1, & \text{satisfactory feedback} \\ 0, & \text{unsatisfactory feedback} \end{cases}$$

$$D_{ij} = \frac{\sum_{l=1}^{n} tlf_{ij}\, f_{ij}}{\sum_{l=1}^{n} tlf_{ij}} \qquad (1)$$

2. **Cooperativeness:** It assesses if the trustee entity engages in social cooperation with the trustor. It is presumed that entities sharing mutual friends tend to cooperate. In the context of Social Internet of Things (SIoT), an object's cooperativeness can be predicted based on its social connections. Each device or object maintains a list of friends who are likely to cooperate. The degree of cooperativeness, denoted as C_{ij}, between two nodes, i and j, is determined by considering their social friendship.The equation 2 quantifies the level of cooperation observed by node i in node j. It relies on direct observations made by node i regarding the cooperative behavior of node j.

$$C_{ii} = \frac{\text{nodes(i)} \cap \text{nodes(j)}}{\text{nodes(i)} \cup \text{nodes(j)}} \qquad (2)$$

3. **Centrality:** Here, G_{ij} indicates the centrality of object j concerning object i, signifying the significance of object j specifically for object i, rather than for the

entire network. This metric serves as a deterrent against malicious nodes that might attempt to establish numerous connections within the network. In equation 3Cij shows common friends among i and j, F_i is friends of i.

$$\left|G_{ij}\right| = \left|C_{ij}\right| / (F_i - 1) \quad (3)$$

5. Machine Learning Approaches

Artificial Intelligence approach is increasingly utilized for trust management in various domains where assessing, maintaining, and ensuring trust is crucial. We can say that, these are programs which are capable of learning from hidden patterns of data, making predictions, and refining their ability based on experience. For instance, simple linear regression can predict outcomes, as seen in stock market forecasting, while the KNN algorithm is adapt at classification tasks. Machine learning, is classified into various types as shown in figure 2 [18].

1. **Supervised Learning**: In supervised learning, models undergo training using labeled datasets, where each input data point is coupled with a corresponding target or output. The objective of this model is to learn the mapping from inputs to outputs, enabling it to make predictions or classifications when faced with new, previously unseen data. Supervised learning includes number of algorithms designed for specific tasks. These encompass linear regression, decision trees, support vector machines, and neural networks.
2. **Unsupervised Learning**: Unsupervised learning is a paradigm in which models are trained on unlabeled data, and the algorithm's aim is to identify inherent patterns, relationships, or structures within the data, all without the need for explicit guidance or labels[19].It encompasses several categories, such as clustering, which involves organizing data into clusters based on similarities, by using methods like k-means clustering.

Here, is some ML approach commonly employed for trust management. Machine learning is a subset of AI that empowers computer systems to enhance their performance in specific tasks by acquiring knowledge from data, all without the need for explicit programming.

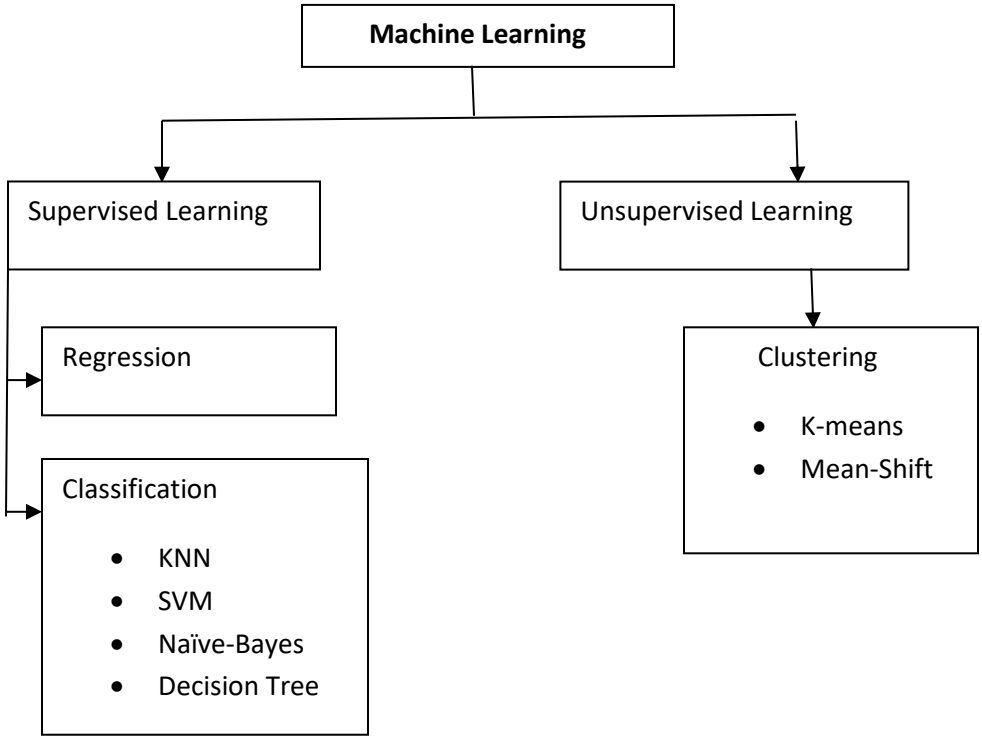

Figure.2. Machine Learning Approaches

1. K-Nearest Neighbor

The K-Nearest Neighbor (K-NN) algorithm represents a fundamental approach in the realm of Machine Learning, based on Supervised Learning.It works by assuming similarity among the new dataset and the existing dataset. This algorithm stores all the available data and classifies a new data point based on the similarity.K-NN serves as an effective tool for both Regression and Classification tasks.It is non-parametric algorithm that means it doesn't make any priori assumptions about the underlying data distribution [20].This algorithm during training phase retains the dataset, and upon the arrival

of new data, it classifies data which is similar to the new data.

Algorithm of K-NN approach is illustrated as fellows [21]:

1. Begin by selecting the desired number of neighbors, denoted as "K".
2. Compute the Euclidean distance between the data point of interest and the K selected neighbors.
3. Choose the K nearest neighbors based on the calculated Euclidean distances.
4. Within this group of K neighbors, tally the occurrences of data points belonging to each category or class.
5. Assign the new data point to the category that boasts the highest number of neighbors within the selected K.

2. Support Vector Machine

The Support Vector Machine (SVM) is a widely used as Supervised learning algorithm that finds utility in both Classification and Regression tasks. But it is most prominently used for Classification task in Machine Learning.The primary objective of the Support Vector Machine algorithm is to establish decision boundary capable of effectively partitioning an n-dimensional space into distinct classes [22]. This enables the straightforward categorization of new data points into their respective categories in subsequent instances. This optimal decision boundary is formally referred to as a "hyperplane".

The SVM, or Support Vector Machine, selects critical data points known as support vectors to construct the hyperplane. These support vectors are the extreme cases pivotal in defining the decision boundary, hence termed as "Support Vector Machine" for this algorithm.

Algorithm of SVM is illustrated below [23]:

1. Ensure that your data is appropriately preprocessed and feature scaled.
2. Choose a kernel function (e.g., linear, polynomialfunction, etc.) that transforms the data into a higher-dimensional space if necessary. The choice of the kernel depends on the problem's characteristics.
3. Initialize the weight vector (w) and bias (b).Set the regularization parameter C, which controls the trade-off between maximizing the margin and minimizing classification errors.
4. Formulate the optimization objective to maximize the margin while minimizing classification errors. This typically involves solving a convex quadratic optimization problem.
5. Use optimization techniques to find the optimal values of w and b that satisfy the optimization objective.
6. Identify the support vectors from the training data.
7. Compute the decision boundary or hyperplane equation using the parameters w and b obtained from the optimization.
8. Compute the decision function: f(x) = w * x + b.Assign the class label based on the sign of f(x). If f(x) > 0, classify as +1; if f(x) < 0, classify as -1.

3. Naïve- Bayes Approach

The Naïve Bayes algorithm is a type of supervised learning technique based on Bayes' theorem, specifically designed for addressing classification problems. Its primary application often lies within text classification, involving datasets with high-dimensional features.The Naïve Bayes Classifier stands out as highly efficient classification algorithm, facilitating the rapid development of machine learning models capable of quick predictions [24].It operates as a probabilistic classifier, relying on the probability of an object to make its prediction.

Algorithm of Naïve-Bayes is illustrated as follows [25]:

1. Collect and preprocess your dataset.
2. Calculate the prior probability of each class:
   P(y = 1): The probability that an example belongs to class 1.
   P(y = 0): The probability that an example belongs to class 0.
3. For each feature in the feature vector x, calculate the occurance of that feature given each class:P(xi|y = 1): The probability of observing feature xi in examples of class 1.
   P(xi|y = 0): The probability of observing feature xi in examples of class 0.
4. Calculate these probabilities based on the occurrences of xi in each class.
5. Given a new data point with feature vector x_new, calculate the posterior probabilities for both classes using Bayes' theorem:
   P(y = 1 | x_new) = (P(x_new|y = 1) * P(y = 1)) / P(x_new)
   P(y = 0 | x_new) = (P(x_new|y = 0) * P(y = 0)) / P(x_new)
6. Assess the performance of your Naive Bayes classifier using appropriate evaluation metrics like accuracy, precision etc depending on the classification problem.
7. Depending on your evaluation results, iterate on the model.

4. Decision Tree

A Decision Tree is a supervised learning approach applicable to both classification and regression problems, although it is commonly preferred for tackling classification tasks. It manifests as a tree-like classifier, wherein internal nodes represent dataset features, branches signify decision rules, and each leaf node corresponds to an outcome [26]. Within a Decision Tree, two fundamental types of nodes exist: Decision Nodes and Leaf Nodes. Decision Nodes are employed to make decisions and comprise multiple branches, while Leaf Nodes denote the final output resulting from these

decisions and lack any additional branching. Decisions or tests within a Decision Tree rely on the characteristics of the provided dataset. This technique offers a graphical representation to explore all potential solutions to a problem contingent on the specified conditions.

Algorithm of Decision Tree is illustrated below [27]:

1. Initiate the tree with a root node, denoted as S, encompassing the entire dataset.
2. Determine the optimal attribute within the dataset using an Attribute Selection Measure (ASM).
3. Partition S into subsets, each containing potential values for the selected best attribute.
4. Establish a decision tree node that encapsulates the chosen best attribute.
5. Recursively build new decision trees using the subsets of the dataset generated in step 3. Continue this iterative process until reaching a point where further classification becomes unfeasible, designating the ultimate node as a leaf node.

5.5 K-Means Clustering

K-Means Clustering represents an Unsupervised Learning technique that organizes an unlabeled dataset into distinct clusters. In this method, the parameter "K" specifies the number of predefined clusters to generate. For instance K=3 leads to three clusters, k=5 leads to five clusters and so on.

Algorithm of K-Means Clustering is illustrated below [28]:

1. Determine the number of clusters (K) that you wish to create.
2. Choose K random points or centroids. These initial centroids may be unrelated to the input dataset.
3. Associate each data point with its nearest centroid, thereby establishing K predefined clusters.
4. Compute the variance within each cluster and position new centroids accordingly.

5. Iterate through the third step, which involves reassigning each data point to the nearest new centroid of its cluster.
6. If any reassignments have transpired, return to step 4; otherwise, proceed to FINISH.

5.6 Mean- Shift Approach

The Mean-shift approach is classified as a clustering technique, distinct from unsupervised learning, where it iteratively assigns data points to clusters by shifting them towards the mode. Consequently, it is often referred to as the Mode-seeking approach. The Mean-shift approach finds practical applications in areas such as image processing and computer vision.Mean-shift clustering stands out as a non-parametric, density-based clustering method employed for cluster identification within datasets. It shines in scenarios where clusters exhibit irregular shapes and aren't distinctly separated by linear boundaries. The fundamental concept involves shifting individual data points toward the mode, or the point with the highest density, within a specified radius. This process iterates until the points converge towards a local density function maximum. These local maxima effectively depict the clusters present in the dataset.

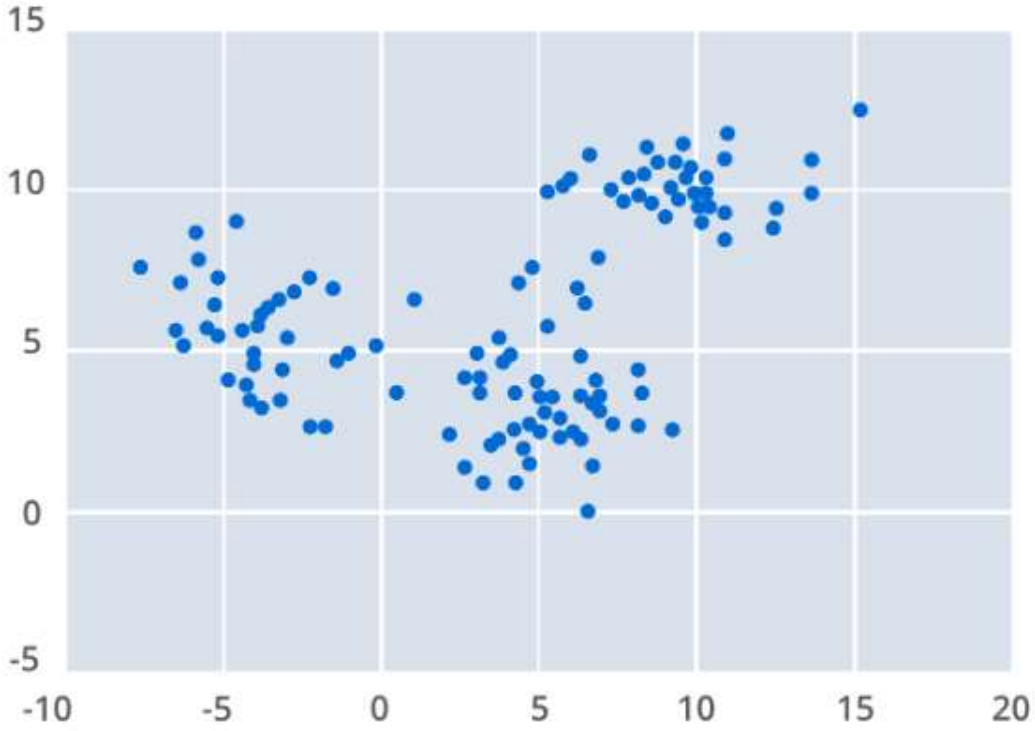

Figure.3. Mean-Shift approach

Algorithm of Mean-shift approach is illustrated below [29]:

1. Begin by initializing the algorithm with data points assigned to individual clusters.
2. Calculate the centroids for each cluster.
3. Update the positions of the centroids based on the data points assigned to their respective clusters.
4. Iterate through the process, moving toward regions of higher data point density.
5. Continue iterating until the centroids reach a position where they no longer exhibit significant movement, indicating convergence.

Approach	Description	Applications
K-Nearest Neighbor	It operates on the principle of presuming a resemblance between the newly acquired dataset and the pre-existing one	Handwriting Detection, Image Recognition, And Video Recognition

Support Vector Machine	This approach is used to create a decision boundary that can efficiently divide search space into an n-dimensional space	Face Detection, Image Classification, Bioinformatics
Naïve-Bayes	This approach functions as a probabilistic classifier, utilizing the probability of an item to inform its prediction	Sentiment Analysis, Spam Filtering, Recommendation Systems
Decision Tree	It is graphical representation that allows for the exploration of all possible solutions to a problem within the defined conditions	Diagnosis of Diseases and Ailments, Retention of Customers
K-Means	K-Means clustering organizes an unlabeled dataset into distinct clusters	Document Clustering, Image Segmentation And Image Compression
Mean-Shift	It involves gradually moving individual data points closer to the mode, which is the data point with the highest density	Image Segmentation, Motion Tracking

Table.1. Summary of ML Approaches

Conclusion

The emergence of Internet of things has allowed virtual objects to communicate with one another and with human being via internet has made them smarter. We have discussed about Internet of Things, trust management architectureis crucial for establishing and maintaining trust among the various IoT components and entities as it involves several key components and principles. It has been observed that assessing trust in IoT systems involves evaluation of trust management mechanisms to ensure their effectiveness and reliability. Key aspects of IoT trust evaluation include: Trust Metrics, Simulation and Testing, Real-world Deployment ,Monitoring and Auditing, User Feedback and Adaptation, Learning, Compliance and Regulations and

Performance Impact.Trust is a crucial concept for aiding entities in decision-making when confronted with uncertainty. In the computation of trust, various trust metrics are employed such as direct trust, cooperativeness and centrality. Machine learning approaches such as KNN, SVM, Naïve-Bayes, Decision Tree, K-Mean and Mean-Shift algorithm. These approaches can be, applied in trust management across different domains to enhance theassessment and maintenance of trust, enabling systems to learn from data and improve its performance over time.

References

[1] Luigi Atzori , Antonio Iera , GiacomoMorabito, and Michele Nitti, 2012, " The Social Internet of Things (SIoT) When social networks meet the Internet of Things: Concept, architecture and network characterization", ComputerNetworks,vol. 56, iss. 16.

[2] Sarma, D. Brock, and K. Ashton,1999, "The networked physical world: proposals for the next generation of computing commerce", automaticidentification. AutoID Center White Paper.

[3] Stergiou, C., Psannis, K.E., Kim, B.-G.,Gupta, B., 2018, "Secure integration of IoT and cloud computing" . Futur.Gener.Comput.Syst. 78(3), 964–975.

[4] Yan, Z., Zhang, P., Vasilakos, A.V., 2014, "A survey on trust management for Internet of Things", J. Netw. Comput. Appl., 42, pp. 120–134.

[5] Chen R, Guo J, Bao F, 2016,"Trust management for soa-based iot and its application to service composition". IEEE Trans ServComput 9(3):482–495.

[6] Jayasinghe U, Truong NB, Lee GM, Um T-W, 2016, "Rpr: a trust computation model for social internet of things". In: Ubiquitous intelligence & computing, advanced and trusted computing, scalable computing and communications, cloud and big data computing,internet of people, and smart world congress.

[7] BernabeJB, Ramos JLH, Gomez AFS, 2016, "Taciot: multidimensional trust-aware access control system for the internet of things". SoftComput 20(5):1763–1779.

[8] Hellaoui H, Bouabdallah A,Koudil M, 2016, " Tas-iot: trust-based adaptive security in the iot". In: Local Computer Networks (LCN), 2016 IEEE 41st Conference.

[9] Kotis, K., Athanasakis, I., &Vouros, G. A., 2018, “Semantically enabling IoT trust to ensure and secure deployment of IoT entities”. International Journal of Internet of Things and Cyber-Assurance, 1(1), 3-21.

[10] Bouazza, H., Said, B., Laallam, F.Z., 2020,“A hybrid IoT services recommender system using social IoT”. J. King Saud UnivComput. Inform. Sci. 8, 206459–206473.

[11] Mehdi Gheisari, Hamid EsmaeiliNajafabadi, Jafar A. Alzubi, JiechaoGao , Guojun Wang, AaqifAfzaalAbbasi, Aniello Castiglione, 2021, "OBPP: An ontology-based framework for privacy-preserving in IoT-based smart city", vol. 123,pp.1-13.

[12] Guo J, Chen R, Tsai JJP, 2017, "A survey of trust computation models for service management in internet of things systems". ComputCommun,97,pp 1–14.

[13] Yan Z, Zhang P, Vasilakos AV, 2014,“A survey on trust management for internet of things”, J NetwComputAppl 42, pp120–134.

[14] Grandinson, T.; Sloman, M.,2000,"A Survey of Trust in Internet Applications", IEEE Communications Surveys.

[15] Lahlou, S.; Langheinrich, M.; Röcker, C.,2005, "Privacy and Trust Issues with Invisible Computers". Communications of ACM, Mar. Vol. 48, No.3, pp. 59-60.

[16] Yan, Z., Chen, Y., Shen, Y., 2014, “PerContRep: a practical reputation system for pervasive content services”, J. Supercomput., 70, (3), pp. 1051–1074.

[17] A. MeenaKowshalya, M.L. Valarmathi, 2017, "Trust management for reliable decision making among social objects in the Social Internet of Thing",IETNetw.,vol 6, iss.4, pp75-80.

[18] Soro, A., Brereton, M., Roe, P.,2018, "Social Internet of Things, 1st edn. Springer.

[19] Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M. and Edwards, D. D., 2003, "Artificial intelligence: a modern approach". Prentice hall Upper Saddle River.

[20] Gou, J. et al.,2019, "A sgeneralised mean distance-based k-nearest neighbor classifier". Expert Syst. Appl. 115,pp. 356–372.

[21] Keller, J. M., Gray, M. R. & Givens, J. A.,1985, "A fuzzy k-nearest neighbor algorithm", IEEE Trans. Syst. Man Cybern. 15,pp. 580–585.

[22] L. U. Yan-Ling, L. I. Lei, MengMeng Zhou, and et al.,2009, "A new fuzzy support vector machine based on mixed kernel function".In ICMLC, pp.526–531.

[23] Yuan Hai Shao, Chun Hua Zhang, Xiao Bo Wang, and Nai Yang Deng,2011, "Improvements on twin support vector machines". IEEE Trans. Neural Netw, 22(6),pp. 962–968.

[24] A. Meehan, C.D. Campos,2015," Averaged extended tree augmented naive classifier". Entropy 17(7),pp. 5085–5100.

[25] J. Karandikar, T. Mcleay, S. Turner, et al.,2015, "Tool wear monitoring using naïve Bayes classifiers". Int. J. Adv. Manuf. Technol. 77(9-12),pp. 1613–1626.

[26] Anyanwu MN, Shiva SG, 2009, “Comparative analysis of serial decision tree classification algorithms”. International Journal of Computer Science and Security,3(3), pp. 230-40.

[27] Drazin S, Montag M, 2012, "Decision tree analysis using weka". Machine Learning-Project II, University of Miami, pp.1-3.

[28] S. Na, L. Xumin and G. Yong, 2010, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics,pp. 63-67.

[29] Seridi-Bou, H.; Sari, T.; Sellami, M, 2005, “A Neural Network for Generating Adaptive Lessons”. J. Comput. Sci., 1, 232–243.