Leveraging Data Mining to Innovate Agricultural Applications

Divya Sai Jaladi[1*], Sandeep Vutla[2]
[1]Senior Lead Application Developer, SCDMV, 10311 Wilson Boulevard, Blythewood, SC 29016, UNITED STATES
[2]Assistant Vice President, Senior-Data Engineer, Chubb, 202 Halls Mill Rd, Whitehouse Station, NJ 08889, UNITED STATES

*Corresponding Author: divyasaij26@gmail.com

ABSTRACT

The WEKA (Waikato Environment for Knowledge Analysis) provides a comprehensive suite of tools and functionalities for applying data mining techniques to large and complex datasets. Designed to support both academic research and industrial applications, WEKA offers a user-friendly graphical interface along with a wide range of machine learning algorithms for tasks such as classification, regression, clustering, association rule mining, and visualization. This paper presents a detailed process model for conducting data analysis using WEKA and highlights the platform's capabilities in supporting each phase of this model—from data preprocessing to model evaluation and deployment. Furthermore, the paper explores how the knowledge models generated by WEKA's data mining algorithms can be seamlessly integrated into larger software systems, thereby enabling the development of intelligent applications. The effectiveness of this approach is demonstrated through a real-world case study in the agricultural domain, specifically focusing on mushroom grading. In this case study, WEKA was utilized to analyze and interpret agricultural data, leading to the creation of a predictive model that can assist in automating the quality assessment of mushrooms based on key attributes. This demonstrates the practicality and scalability of WEKA in supporting domain-specific data-driven decision-making and application development.

Keywords: Machine Learning, Data Mining, Data Analysis, Application Development

Introduction

Data mining is the way to find obscure and conceivably fascinating designs in enormous datasets. The 'mined' data is ordinarily addressed as a model of the semantic construction of the dataset, where the model might be utilized on new information for forecast or characterization. On the other hand, human area specialists may physically analyze the model, looking for partitions that clarify recently misconstrued or obscure attributes of the space under investigation [1].

Our work focuses on A.I. strategies for actuating space models or then again breaking down datasets. Machine Learning calculations furnish models with a grouping/expectation precision equivalent to, for instance, counterfeit neural organizations, however, which are more apparent to people than a neural model.

The WEKA research group has two destinations: to mine data from existing agrarian datasets created by New Zealand researchers and exploration associations; and to perform fundamental exploration in information mining by growing new A.I. calculations. To help these objectives, we have built up an information mining workbench, the WEKA framework, that joins the accompanying devices: a bunch of information pre-preparing schedules, supporting the control of crude information and its change into a proper structure for information mining; include determination devices, valuable for distinguishing superfluous

credits to prohibit from the dataset; classifiers and other information mining calculations, equipped for taking care of straight out and numeric learning undertakings; meta classifiers for upgrading the presentation of grouping information mining calculations (for instance, boosting and packing schedules); test support for checking the near heartiness of numerous enlistment models (for model, schedules estimating characterization exactness, entropy, root-squared mean blunder, cost-delicate characterization, and so on); and benchmarking instruments, for contrasting the family member execution of various learning calculations more than a few datasets [2].

Broad, naturally created documentation of the WEKA source is accessible to control the client through communications with the framework. Prior, Unix-based renditions of the WEKA framework are additionally portrayed. The current form of WEKA is carried out as a set-up of Java class libraries. It is uninhibitedly accessible on the World-Wide Web (http://www.cs.waikato.ac.nz/~ml/weka). The product goes with content on information mining which records and completely clarifies all the information mining calculations joined in WEKA. Application programs composed utilizing the WEKA class libraries can be run on any P.C. with a WWW program.

The overall interaction of information mining is portrayed in Section 2. Segment 3 depicts the explicit A.I. calculations addressed in the current rendition of WEKA, and Area 4 describes WEKA instruments for supporting the information mining measure model. A case study outlining information mining utilizing WEKA shows up in Section 5, and Section 6 presents our ends [3].

Data Mining – Process Model

We have dissected more than 50 certifiable informational indexes throughout this task; research organizations in New Zealand give especially horticultural informational collections. From this experience, we have built up a cycle model for applying information mining methods to information to fuse the actuated area data into a product module (Figure 1). The central issues of this model are:

- A two-path communication between the supplier of the information and the information mining master. Both works together to change the crude information into the last information set(s) contribution to the machine learning calculations — with the space master giving data about information semantics and 'legitimate' changes that can be applied to the information. The information mining master managing the cycle to improve the understandability and exactness of the outcomes.

- An iterative methodology. Data mining is an exploratory interaction. For the most part, it takes a few cycles through the interaction model to track down a decent "fit" between a portrayal of the information and an information mining calculation. Likewise, particular property mixes that go through various plans can create uncontrollably extraordinary information models. However, the visionary precision of the outcomes might be the same. These elective

perspectives may give significant experiences into designs covering various subsets of the information.

In the model introduced in Figure 1, movement streams a clockwise way. In the pre-processing stage, the crude information is initially addressed as a solitary table, as needed by the information mining calculations remembered for WEKA. This table is converted into the ARFF design, a characteristic/esteem table portrayal that retains header data for the credits' information types. Likewise, the information may require extensive 'purging' to eliminate anomalies, handle missing qualities, recognize mistaken attributes, etc [4].

Now the information supplier (space master) and the information mining master team up to change the scrubbed information into a structure that will deliver an intelligible, precise information model at the point when prepared by an information mining calculation. These two experts may, for instance, conjecture that at least one credits are unimportant and put away these unessential sections. Properties might be controlled numerically, for example, to change overall segments containing temperature estimations to a typical scale, standardize values in a given area, or consolidate at least two elements into a solitary determining quality.

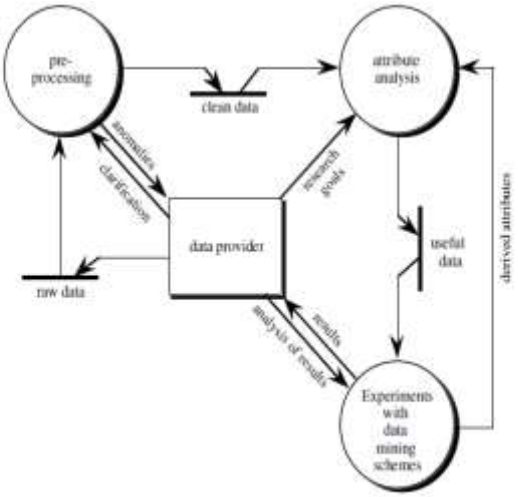

Figure 1. Process model for a machine learning application (data flow diagram)

The information mining plans then prepare at least one form of the scrubbed information. The area master figures out what segments of the yield are adequately novel or intriguing to warrant further investigation and what components address everyday information for that field. The information mining master deciphers the calculations' yield and offers guidance on additional trials that could be run with this information [5].

Data Mining Techniques

The current variant of WEKA contains twelve learning plans: ten classifiers, a grouping calculation, and an affiliation rule student. The product design is adequately adaptable to allow other learning plans and different sorts of learning plans to be opened into WEKA. In this part, we depict the kinds of discoveries that WEKA at present backings.

A. Classifiers

The yield from this sort of learning plan is, in a real sense, a classifier—ordinarily in the structure of a choice tree or set of decides that can be utilized to anticipate the grouping of another information occasion. One property in the information table is assigned as the classification or class for expectation; the remainder of the ascribes may show up in the "if" bits of the guidelines (or the non-leaf hubs of the choice tree) [6].

The most simple learning plan in WEKA is ZeroR (Table 1). This plan models the dataset with a solitary standard. Given another information thing for order, ZeroR predicts the most incessant classification esteem in the preparation information for issues with ostensible class esteem or then again indicates the regular class an incentive for numeric expectation issues. ZeroR helps produce a benchmark execution that other learning plans are contrasted. In a few datasets, it is feasible for other learning plans to initiate models that perform on new information than ZeroR—a marker of generous overfitting.

weka.classifiers.ZeroR
weka.classifiers.OneR
weka.classifiers.NaiveBayes
weka.classifiers.DecisionTable
weka.classifiers.Ibk
weka.classifiers.j48.J48
weka.classifiers.j48.PART
weka.classifiers.SMO
weka.classifiers.LinearRegression
weka.classifiers.m5.M5Prime
weka.classifiers.LWR
weka.classifiers.DecisionStump

Table 1: The basic learning schemes in Weka

The next plan, OneR, creates straightforward principles dependent on a solitary trait. OneR is additionally valuable in creating a benchmark for order execution. Indeed, this calculation was found to proceed just as more modern analyses over many norms A.I. test datasets (Holte, 1993)! It gives the idea that in any event, part of the justification for this outcome is that large numbers of the standard test information bases typify extremely straightforward hidden connections in the information. Certifiable data sets may contain organized data about space, and these direct connections can be closely distinguished and addressed by OneR [7].

NaiveBayes carries out a Naïve Bayesian classifier, which produces probabilistic rules. That is, when given another information thing, the NaiveBayes model demonstrates the likelihood that this thing has a place with every one of the conceivable class classifications. The Bayesian classifier is 'innocent' as in credits are treated like autonomous, and each characteristic contributes similarly to the model. If unessential ascribes are remembered for

the dataset, those credits will slant the model. NaiveBayes, as OneR, can give shockingly great outcomes on numerous natural world datasets regardless of its straightforwardness [8].

Decision Table sums up the dataset with a 'choice table.' A choice table contains a similar number of properties as the first dataset in its most straightforward expression. Another information thing has relegated a classification by discovering the line in the choice table that coordinates with the non-class upsides of the information thing. This execution utilizes the covering technique to track down a decent subset of properties for consideration in the table. By killing ascribes that contribute pretty much nothing or nothing to a dataset model, the calculation decreases the probability of over-fitting and makes a more modest, more dense choice table.

Instance-based learning plans make a model by just putting away the dataset. Another information thing is ordered by contrasting it and these 'retained' information things, utilizing a distance metric. Then again, the more significant part class of the k closest information things might be chosen, or for numeric credits, the distance-weighted normal of the k nearest items might be allocated. IBk is an execution of the k-closest neighbor's classifier. The quantity of closest neighbors (k) can be set physically or decided naturally utilizing cross-approval.

j48 is an execution of C4.5 discharge 8, a standard calculation that is generally utilized for reasonable A.I. This execution produces choice tree models. Part is a later plan for creating sets of rules called "choice records," which are requested arrangements of regulations. Another information thing is contrasted with each standard in the rundown this way. Also, the item is allocated the class of the central coordinating with rule (a default is applied to assume no authority effectively corresponds). This calculation works by framing pruned fractional choice trees (fabricated utilizing C4.5's heuristics) and quickly changing over them into a relating rule [9].

SMO executes the "successive negligible improvement" calculation for support vector machines (SVMs), which are a significant new worldview in AI. SVMs have seen huge applications in learning models for text classification. While SMO is probably the quickest strategy for learning SVMs, it is regularly delayed to join to an answer—especially with boisterous information.

WEKA contains three techniques for the numeric forecast. The least difficult is Linear Regression. LWR executes a more refined learning plan for numeric prediction, utilizing privately weighted relapse. M5Prime is an accurate reproduction of Quinlan's M5 model tree inducer. While choice trees were intended for appointing ostensible classifications, this portrayal can be reached out to numeric forecast by changing the leaf hubs of the tree to contain a numeric worth which is the normal of the relative multitude of dataset's qualities that the leaf applies [10-13].

At last, **Decision Stump** fabricates clear double choice "stumps" (1-level choice trees) for both numeric and ostensible order issues. It adapts to missing qualities by expanding the third branch from the stump—all in all, by treating "missing" as separate trait esteem.

Decision Stump is primarily utilized related to the Logit Boost boosting strategy, examined in the following area.

B. Meta-Classifiers

Late advancements in computational learning hypothesis have prompted strategies that improve the exhibition or broaden the abilities of these essential learning plans. We call these execution enhancers "meta-learning plans" or "meta-classifiers" since they work on the yield of different students. Rather than utilizing a solitary classifier to make forecasts, why not mastermind a classifier council to decide on the grouping and case? This is the essential thought behind joining various models to frame a troupe or meta classifier [14-17].

Two of the most noticeable techniques for building gathering classifiers are boosting and packing. Usually, these classifiers can increment prescient execution over a solitary classifier. In any case, the cost for this expansion in performance is that it is by and mainly impractical to comprehend what is behind the improved choice making.

Both stowing and boosting vote on arrangements utilizing a weighted vote—each model in the outfit predicts a class and appoints a certainty worth to the forecast. These values are added, and the course with the enormous worth (most certainty) is picked.

The two techniques infer their assortments of models of the information in very various manners. Packing works by building separate models of the preparation information utilizing an inspecting method that erases a few occasions and reproduces others. In this manner, singular models are constructed independently with a crisp preparing set at every emphasis (the quantity of cycles decides the number of models built for the troupe).

Algorithmically:

Let n be the number of instances in the training data
For each of t iterations do
Randomly sample n instances (using deletion and replication)
Apply a learning technique to build a model from the sample
Store the model
End

Like sacking, boosting is iterative, yet as opposed to examining crisp preparing information, and each new model is affected by the exhibition of those fabricated beforehand. Occurrences that are erroneously characterized in the past emphasize they are advanced and those effectively ordered are consigned. The critical thought is to weigh the cases and utilize a learning calculation to consider these loads while developing its models. At first, the loads are indeed even, and a model is developed. The cases effectively ordered by this model are given less weight, so the inaccurately arranged issues will have more "significance" in the next cycle.

Algorithmically:

Assign equal weight to all instances
For each of t iterations do
Apply a learning technique to build a model from the weighted instances
and store the resulting model
Down-weight each instance correctly classified by the model
End

The AdaBoost.M1 boosting calculation gives the client control ridiculous cycles performed. Another boosting methodology is carried out by LogitBoost, which fits issues including two-class circumstances—for instance, the SMO class from a higher place. It is essential to change the multi-class case into a few two-class ones and join the outcomes. The MultiClassClassifier boosting method does precisely that.

C. Clustering

Clustering techniques don't create prescient principles for a specific class but instead attempt to track down the characteristic groupings (or "bunches") in the dataset. This procedure is regularly utilized in an exploratory style to produce speculations about the connections between information occasions. A subsequent learning stage frequently trails bunching. A classifier is used to actuate a standard set or choice tree that distributes each case in the dataset to the group relegated by the bunching calculation. These classifier-produced 'bunch portrayals' would then be analyzed to acquire a semantic comprehension of the groups.

WEKA incorporates an execution of the E.M. grouping calculation. This calculation makes the presumption, usual to other grouping calculations, that the credits in the dataset address free arbitrary factors. E.M. allows an occurrence to have a home with more than one group, a valuable augmentation that, by and by, can uphold more adaptable and then some 'fluffy' depictions of the verifiable construction of the dataset [18-22].

D. Affiliation rules

WEKA contains the execution of the Apriori calculation for creating affiliation administers, a sort of learning plan ordinarily utilized in "market bin examination" (MBA). MBA calculations have, as of late, seen boundless use in breaking down customer buying designs—explicitly, in identifying items that are much of the time bought together. These calculations were created based on the tremendous surge of exchange information delivered by scanner tag-based buying/requesting frameworks. The business world immediately perceived this information as having gigantic possible worth in advertising, yet, conventional information examination procedures couldn't adapt to the size of the speculation space that these datasets cause.

For this kind of examination, information is coherently coordinated into "bins" (typically records in which the things bought by a given shopper at a given time are assembled). MBA calculations, for example, Apriori find "affiliation controls" that distinguish examples of buys, to such an extent that the presence of one thing in a container will suggest the company

of one or more extra things. A hypothetical illustration of such a standard may be that customers who buy toothpaste are likewise liable to purchase bananas on a similar excursion to the supermarket. This outcome would then be utilized to propose blends of items for extraordinary advancements or deals, devise a more compelling store format, and understand brand reliability and cobranding.

WEKA Tools

Notwithstanding the learning calculations examined above, WEKA additionally gives apparatuses for pre-processing information and for looking at the presentation of various learning calculations.

1. Dataset Pre-Processing

WEKA's pre-handling ability is epitomized in a broad arrangement of schedules, called channels, that empower information to be handled at the case and trait esteem levels. These channels have a standard order line interface with a bunch of typical order line alternatives.

A large number of the filter algorithms give offices to the general control of traits—for a model, to embed and erase credits from the dataset. While trying different things with learning plans to improve an information mining application (Section 2), one of the most basic exercises includes building models with various subsets of the complete characteristic set. WEKA gives three-component determination frameworks to help in picking ascribes for consideration in a trial: a privately delivered connection-based strategy.

At times, it may be valuable to apply a change capacity to a whole section in the dataset—for instance, to standardize each worth in a trait. For ostensible ascribes, it might be beneficial to address a multi-class marker as a two-class property; on the other hand, the number of classes might be decreased by consolidating two upsides ostensible trait into a single worth. Channels are given to help these changes, which might be applied to non-class credits also. Since some learning plans (e.g., SMO) can deal with parallel ascribes, a channel is accessible that changes a multi-esteemed ostensible quality into a parallel esteemed quality. Since numerous calculations can't deal with genuine decent credits, a discretization channel is given. It can perform solo discretization or administered discretization.

Missing qualities happen regularly in real-world datasets and are hard for some information mining calculations to deal. Undoubtedly, most calculations discard lines of information containing a missing worth—which can, in outrageous cases, diminish the measure of useable data in a preparing set to the point that a solid model can't be shaped. One procedure for managing with missing qualities is to universally supplant them with assessed values before a learning plot is applied; WEKA gives a channel that substitutes the mean (for numeric credits) or the mode (for ostensible ascribes) for each missing worth.

The presence of exceptions in a dataset may truly slant a model; a channel can eliminate anomalies by erasing all occurrences that display one specific arrangement of ostensible trait values or a numeric worth under a given edge.

2. Benchmarking algorithm performance

One of the vital parts of the WKA suite is the office it gives to assess learning plots reliably. For instance, a scientist can make an "alliance table" summing up the relative presence of a few plans over various datasets. Table 2 delineates the consequences of applying ten classifiers to 37 datasets from the UCI store (Blake and Merz, 1998), a massive vault of benchmark datasets for the A.I. research local area.

W-L	Wins	Loss	Scheme
208	254	46	LogitBoost -I 100 Decision Stump
155	230	75	LogitBoost -I 10 Decision Stump
132	214	82	AdaBoostM1 Decision Trees
118	209	91	Naïve Bayes
62	183	121	Decision Trees
14	168	154	IBk Instance-based learner
-65	120	185	AdaBoostM1 Decision Stump
-140	90	230	OneR—Simple Rule learner
-166	77	243	Decision Stump
-195	9	204	ZeroR

Table 2: Ranking schemes

Section 2, Wins, is the number of datasets for which the plan performed better (at the 95% certainty level) than another plan. Misfortune is the number of datasets for which a plan served fundamentally more regrettable than another plan. W-L is the contrast among wins and misfortunes to give a general score. Doubtlessly, for these 37 test sets, that Logit boosting short stumps for 10 or 100 cycles is the best generally speaking strategy among the plans accessible in WEKA.

Case Study: Mushroom Grading

The information mining measure model (Section 2) has helped focus the investigation of real-world datasets, utilizing the WEKA examination instruments (Section 4) and the WEKA learning plans (Area 3). In this part, we portray one such application: a grouping framework for arranging mushrooms by grade.

The objective of this venture was to prompt an ordered framework fit for arranging mushrooms into quality evaluations and accomplishing an exactness like that achieved by human assessors. This exploration was done in a joint effort with individuals from a neighborhood rural exploration association, who took the piece of the information supplier/area master in the process model (Section 2).

For this situation, the information pre-handling stage included not just the purifying of crude information but also the development of a test dataset as a team with the farming specialists. This dataset contains portrayals of 282 mushrooms. The credits included both goal measures (weight, immovability, level of cap opening) and emotional measures (Likert scale assessments of soil level, tail harm wounding, shrink, bacterial smudge, and P. gingeri).

Three controllers autonomously reviewed the mushrooms utilizing the three expansive business grades (first, second, and third grade).

Also, computerized pictures were caught for the 282 mushrooms. These pictures were broke down to give an extra 60 shot based credits: recurrence canister esteems (0-4) from the investigation of Red, Green, and Blue (R, G, B) and Hue, Saturation and Value (H, S, V) histograms for top (t) and base (b) pictures of the example mushrooms.

The wrapping method, related to demonstrating building utilizing the J4.8 classifier, proved valuable in dispensing with a considerable lot of these 68 ascribes (relating to the "property examination" what's more, "tries different things with A.I. plans" cycle in the process model, Figure 1). Utilizing J4.8 and covering search, a different model was created for the three assessors. The models created recommended that every controller used various mixes of credits when relegating evaluations to mushrooms. Every one of the visionary models utilized credits from top and base pictures. Just Inspector 2 utilized load for the order of mushrooms into three evaluations. The emotional estimations (soil, tail harm, wounding, wither, bacterial smudge, and P. gingerii) didn't expand the precision of any of the expectation models, as were disposed of by the covering strategy. Finally, the models each joined somewhere in the range of four and seven ascribes—a critical decrease from the unique 68! The normal exactness of the models was contrasted well and that of the by and large, worthy.

These outcomes demonstrate that outwardly based credits, which can be consequently separated from digitized pictures, are adequate for the acceptable partition of mushrooms into three expansive quality groups (where 'great' is estimated in contrast with human evaluating norms). The abstract ascribes, regularly accepted to assume a significant part in evaluating, are extra to the undertaking. This astounding piece of 'mined' data echoes the finishes of an exemplary A.I. paper. This incited many rules for diagnosing soybean infections that were strikingly not at all like well-qualified conclusions on the proper conclusion technique—however, which were exact to such an extent that one master received the found standards instead of his own!

A more precise, however humanly garbled, model was made utilizing boosting. As examined in Section 3, upgrading is a "discovery" approach for delivering a gathering of models that, on the whole, accomplish higher exactness. In one trial, 50 models were developed and appraised with AdaBoost. When making an expectation for new information, the singular models each have a vote relative to their precision on the preparation information—subsequently, the higher precision in evaluating mushrooms accomplished by this model, yet at the cost of its lucidness.

Conclusion

As shown by the contextual analysis introduced in Section 5, data 'mined' from information can give experiences into the area being examined that oppose a field's astuteness. Finding these astonishing or surprising bits of the model can be the concentration for an information mining investigation. The goal is that the outcomes can be applied back in the area from which the information was drawn. For this situation, the results demonstrate that the

emotional ascribes for mushroom evaluating may not be valuable practically speaking, and thus maybe they need not be estimated or recorded. Measures dependent on the credits found in the J4.8 models may help grow more target guidelines for quality grouping and market valuing for mushrooms.

In other data mining applications, the objective may be to utilize a model presciently to give mechanized grouping new occurrences. In these applications, the learning part will probably be a little piece of a lot more powerful programming framework. Since WEKA learning plans are available from different projects, a learning module can be opened into a more extensive framework with additional programming.

Figure 2, for instance, shows a WEKA applet dependent on a J4.8 mushroom evaluating model, as depicted in Section 5. Picture preparing an image of a mushroom cap (at left in Figure 2) gives information to the model to separate between A, B, and C evaluation mushrooms. Various models, as produced from WEKA, can be handily subbed into the applet as wanted.

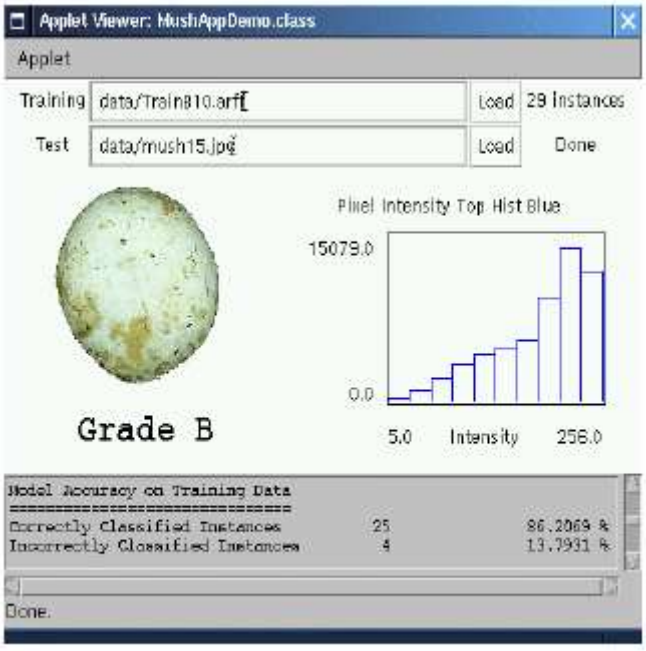

Figure 2: Mushroom grading applet

As the innovation of A.I. proceeds to create and develop, learning calculations should be brought to the work areas of individuals who work with information and comprehend the application space from which it emerges. It is vital to get the calculations out of the lab and into the workplace of the individuals who can utilize them. WEKA is a considerable advance in the exchange of A.I. innovation into the working environment.

References

[1] Pasham, S.D. (2017) AI-Driven Cloud Cost Optimization for Small and Medium Enterprises (SMEs). The Computertech. 1-24.

[2] Goriparthi, R.G. (2020) Neural Network-Based Predictive Models for Climate Change Impact Assessment. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence. 11(1): 421-421

[3] Pasham, S.D. (2018) Dynamic Resource Provisioning in Cloud Environments Using Predictive Analytics. The Computertech. 1-28.

[4] Manduva, V.C. (2020) AI-Powered Edge Computing for Environmental Monitoring: A Cloud-Integrated Approach. The Computertech. 50-73.

[5] Agarwal, A. V., Verma, N., & Kumar, S. (2018). Intelligent Decision Making Real-Time Automated System for Toll Payments. In Proceedings of International Conference on Recent Advancement on Computer and Communication: ICRAC 2017 (pp. 223-232). Springer Singapore.

[6] Ravichandran, N., Inaganti, A. C., Muppalaneni, R., & Nersu, S. R. K. (2020). AI-Driven Self-Healing IT Systems: Automating Incident Detection and Resolution in Cloud Environments. Artificial Intelligence and Machine Learning Review, 1(4), 1-11.

[7] Ravichandran, N., Inaganti, A. C., Muppalaneni, R., & Nersu, S. R. K. (2020). AI-Powered Workflow Optimization in IT Service Management: Enhancing Efficiency and Security. Artificial Intelligence and Machine Learning Review, 1(3), 10-26.

[8] Pasham, S.D. (2020) Fault-Tolerant Distributed Computing for Real-Time Applications in Critical Systems. The Computertech. 1-29.

[9] Inaganti, A. C., Sundaramurthy, S. K., Ravichandran, N., & Muppalaneni, R. (2020). Zero Trust to Intelligent Workflows: Redefining Enterprise Security and Operations with AI. Artificial Intelligence and Machine Learning Review, 1(4), 12-24.

[10] Inaganti, A. C., Sundaramurthy, S. K., Ravichandran, N., & Muppalaneni, R. (2020). Cross-Functional Intelligence: Leveraging AI for Unified Identity, Service, and Talent Management. Artificial Intelligence and Machine Learning Review, 1(4), 25-36.

[11] Nersu, S. R. K., Kathram, S. R., & Mandaloju, N. (2020). Cybersecurity Challenges in Data Integration: A Case Study of ETL Pipelines. Revista de Inteligencia Artificial en Medicina, 11(1), 422-439.

[12] Srinivas, N., Mandaloju, N., & Nadimpalli, S. V. (2020). Cross-Platform Application Testing: AI-Driven Automation Strategies. Artificial Intelligence and Machine Learning Review, 1(1), 8-17.

[13] Mandaloju, N., Srinivas, N., & Nadimpalli, S. V. (2020). Machine Learning for Ensuring Data Integrity in Salesforce Applications. Artificial Intelligence and Machine Learning Review, 1(2), 9-21.

[14] Mahmud, U., Alam, K., Mostakim, M. A., & Khan, M. S. I. (2018). AI-driven micro solar power grid systems for remote communities: Enhancing renewable energy

efficiency and reducing carbon emissions. Distributed Learning and Broad Applications in Scientific Research, 4.

[15] Pasham, S.D. (2019) Energy-Efficient Task Scheduling in Distributed Edge Networks Using Reinforcement Learning. The Computertech. 1-23.

[16] Manduva, V.C. (2020) How Artificial Intelligence Is Transformation Cloud Computing: Unlocking Possibilities for Businesses. International Journal of Modern Computing. 3(1): 1-22.

[17] Alam, K., Mostakim, M. A., & Khan, M. S. I. (2017). Design and Optimization of MicroSolar Grid for Off-Grid Rural Communities. Distributed Learning and Broad Applications in Scientific Research, 3.

[18] Agarwal, A. V., & Kumar, S. (2017, November). Unsupervised data responsive based monitoring of fields. In 2017 International Conference on Inventive Computing and Informatics (ICICI) (pp. 184-188). IEEE.

[19] Agarwal, A. V., Verma, N., Saha, S., & Kumar, S. (2018). Dynamic Detection and Prevention of Denial of Service and Peer Attacks with IPAddress Processing. Recent Findings in Intelligent Computing Techniques: Proceedings of the 5th ICACNI 2017, Volume 1, 707, 139.

[20] Manduva, V.C. (2020) The Convergence of Artificial Intelligence, Cloud Computing, and Edge Computing: Transforming the Tech Landscape. The Computertech. 1-24.

[21] Agarwal, A. V., & Kumar, S. (2017, October). Intelligent multi-level mechanism of secure data handling of vehicular information for post-accident protocols. In 2017 2nd International Conference on Communication and Electronics Systems (ICCES) (pp. 902-906). IEEE.

[22] Goriparthi, R.G. (2020) AI-Driven Automation of Software Testing and Debugging in Agile Development. Revista de Inteligencia Artificial en Medicina. 11(1): 402-421.