

DOI: 10.36910/6775-2524-0560-2020-39-29

УДК: 004.932

Міскевич Оксана Іванівна, асистент

<https://orcid.org/0000-0002-5009-2391>

Багнюк Наталія Володимирівна, к.т.н., доцент

<https://orcid.org/0000-0002-7120-5455>

Хрестинець Наталія Анатоліївна, ст.викладач

<https://orcid.org/0000-0002-4836-7632>

Марчевська Ольга Романівна, студентка

Луцький національний технічний університет

АВТОМАТИЗАЦІЯ ВИЯВЛЕННЯ ДЕФЕКТІВ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

Міскевич О. І., Багнюк Н. В., Хрестинець Н.А., Марчевська О.Р. Автоматизація виявлення дефектної продукції методами машинного навчання. Представлено основні методи машинного навчання для розпізнавання дефектів у сферах виробництва різнопланової продукції. Детально розглянуто застосування підходів глибокого навчання і комп'ютерного бачення для розпізнавання недоліків продукції апаратного забезпечення, одиночних та комплексних алгоритмів машинного навчання для контролю якості програмного забезпечення.

Ключові слова: автоматизація виробництва, виявлення дефектів, CNN, Machine Learning, Deep Learning, Computer Vision, Quality Assurance.

Міскевич О. И., Багнюк Н. В., Хрестинец Н.А., Марчевская О.Р. Автоматизация выявления дефектной продукции методами машинного обучения. Представлены основные методы машинного обучения для распознавания дефектов в областях производства разнородной продукции. Подробно рассмотрено применение подходов глубокого обучения и компьютерного видения для распознавания недостатков продукции аппаратного обеспечения, одиночных и комплексных алгоритмов машинного обучения для контроля качества программного обеспечения.

Ключевые слова: автоматизация производства, выявления дефектов, CNN, Machine Learning, Deep Learning, Computer Vision, Quality Assurance.

Miskevych O, Bahniuk N., Khrystinets N., Marchevska O. Automation of defective products detection by machine learning methods. The basic methods of machine learning for the detection of defects in the fields of production of various products are presented. The use of deep learning and computer vision approaches to identify hardware faults, single and complex machine learning algorithms for quality control of software are discussed in detail.

Keywords: production automation, defect detection, CNN, Machine Learning, Deep Learning, Computer Vision, Quality Assurance.

Постановка наукової проблеми: Залежно від сфери, виробничий процес (далі ВП) може бути складним та супроводжуватись появою помилок. Хоча ВП відрізняється один від одного залежно від сфери впровадження, при виконанні кожного з них прагнуть досягнення подібних цілей, що стосуються загальної ефективності обладнання (overall equipment effectiveness – OEE) та загальної ефективності лінії (overall line efficiency – OLE). OEE розраховується як добуток між показниками наявності, продуктивності та якості. А значення OLE – середньозважене OEE кожної машини на виробничій лінії, а отже важливість якості як основної виробничої метрики – незаперечна.

Аналіз досліджень. Серед всіх сфер застосування машинного навчання (Machine Learning, далі ML) для контролю якості та надійності виділяють дві найважливіші [1].

Прогнозуючий контроль якості. Контроль якості – вбудований у виробництво процес, де дефектні вироби відсіюються від решти продукції якомога раніше. Під час того, як контроль якості забезпечує виявлення браку та попереджає появу дефектного продукту на ринку, сам процес виявлення головної причини, що призвела до появи дефектних виробів вимагає великих затрат часу і часто вимагає залучення багатьох дисциплін – оптимізації процесу, гарантій якості, механіки та електроніки тощо.

Наприклад, харчове виробництво вимагає жорстких мір при контролі якості на кожному кроці ВП для гарантування безпечності їжі. Good Manufacturing Practices (GMP), Hazard Analysis Critical Control Point (НАССР), Hazard Analysis Risk-based Preventive Controls (HARPC), Codex Alimentarius, та ISO 22000 – лише декілька регуляторів та стандартів, розроблених для цієї мети.

Із впровадженням ML, обладнання та дані про виріб можуть повністю відслідковуватись впродовж усього процесу виготовлення для передбачення втрат якості до того, як такі з'являться. Команди управління та якості попереджаються про точні причини очікуваних помилок, що і є основою Quality 4.0 [2].

Інтегрування машинного навчання в процес управління якістю часто відносять до прогнозування якості, що знижує власне проблеми якості та відходи, скорочує виробничі затрати,

мінімізує відкликання продукції та захищає репутацію бренда. Повідомляють також, що ML може підвищити виявлення браку до 90% при цьому зменшуючи час на знаходження причини його появи від днів до хвилин.

Прогнозуюче технічне обслуговування. Відповідно до останніх досліджень, незаплановані простої промислових машин у виробництві оцінюють у \$50 мільярдів щорічно. А 42% простоїв становлять відмови устаткування [3]. Прогнозуюче технічне обслуговування – моніторинг стану обладнання та передбачення необхідності обслуговування на основі ведучих показників появи дефектів.

Прогресуючі техніки ML все більше застосовують для оптимізації результатів прогнозуючого обслуговування. Такі техніки включають моделювання відповідного процесу розробки на виробничій лінії, а тоді застосування найбільш умісного алгоритму ML в контексті ВП та специфіки продукції, що виготовляється. При прогнозуючому обслуговуванні алгоритми ML не повинні отримувати інформацію для навчання з попередньо встановленим пороговим значенням, а їх необхідно «натренувати» для розпізнавання шаблонів та аномалій на прикладах звичайної поведінки та історії помилок. Хоча перед першим застосуванням алгоритми все одно тренують, проте з часом вони самооптимізуються з мінімальним втручанням людини при цьому. Завдяки цьому фабрики передбачають та попереджають простої здійснюючи правильні та вчасні дії.

Автори статті [4] П. Дека та Р. Міттал описують автоматизацію у виробництві. Як відомо, на сьогоднішній день підвищення рівня автоматизації у виробництві також залежить від автоматизації перевірки якості матеріалів з невеликим втручанням людини. Основний напрям – досягнення рівня точності людини або більша якість автоматизованої перевірки. Такий напрям надає користувачу можливість використання деяких аспектів DL та відображає потребу в оптимізації всього циклу (алгоритми, структура виводу, апаратні прискорювачі) для отримання оптимальної продуктивності.

Виклад основного матеріалу й обґрунтування отриманих результатів. Для досягнення виробничих стандартів, інспектори з якості на виробництвах перевіряють якість зазвичай після виготовлення продукції – це ручна робота, що вимагає великих затрат часу, а відхилений виріб призводить до підвищення втрат заводських потужностей, виробничих матеріалів та праці. Відповідно до сучасного тренду штучного інтелекту, заводи та фабрики шукають шляхи впровадження глибокого навчання на основі технологій комп'ютерного бачення (Computer Vision, далі CV) у сам цикл виробництва для автоматизації перевірки якості матеріалів.

Глибоке навчання (Deep learning, далі DL) – сфера вивчення глибоких структур та неструктурованих відображень даних, зростаючий напрям у ML для отримання кращих результатів, при даних великих розмірів та складності.

Згорткові нейронні мережі (Convolution neural Network, далі CNN) – клас глибоких нейронних мереж, що зазвичай використовують для аналізу зображень. Згорткові шари додають згорткові операції для початкової передачі результату до наступного шару. Згорткові операції є рішенням проблеми великої кількості ознак, адже ці операції скорочують число вільних атрибутів, дозволяючи мережі бути глибшою при меншій кількості параметрів.

CNN складають різні типи шарів:

- Згортковий (створює карту атрибутів для передбачення класових ймовірностей для кожного атрибута, застосовуючи фільтр, що сканує все зображення по декілька пікселів за раз);
- Об'єднуючий (зменшення вибірки) (зменшує масштаби інформації, згенерованої згортковим шаром для кожного атрибута та утримує найбільш суттєву інформацію);
- Повністю з'єднаний вхідний ("сплющує" виходи згенеровані попереднім шаром для перетворення їх в один вектор, що може бути використаний як вхідний для наступного шару);
- Повністю з'єднаний (додає ваги на входи, що створені за допомогою аналізу ознак для прогнозування точної мітки);
- Повністю з'єднаний вихідний (створює фінальні ймовірності для класифікації екземпляра).

Є дві головних переваги використання згорткових шарів порівняно з повністю з'єднаними – розподіл параметрів та розрідження зв'язків [5]. Згорткові нейронні мережі здійснюють пошук шаблонів у зображенні. Зображення згортається в меншу матрицю і ця згортка використовується для пошуку шаблонів у зображенні. Декілька перших шарів можуть визначати лінії, кути, краї тощо. Далі ці шаблони передаються в глибші шари нейронної мережі для розпізнавання більш складних ознак. Така властивість CNN чудово спрацьовує при визначенні об'єктів на зображенні.

Наприклад, для задачі розпізнавання пошкоджень на плиті певного матеріалу (рис. 1) застосовано два згорткових шари, повністю зв'язаний (ReLU) та об'єднуючий (Max Pooling).

Очевидно, що архітектура CNN фокусується на пошкоджених блоках та їх поширенні по всій площині:

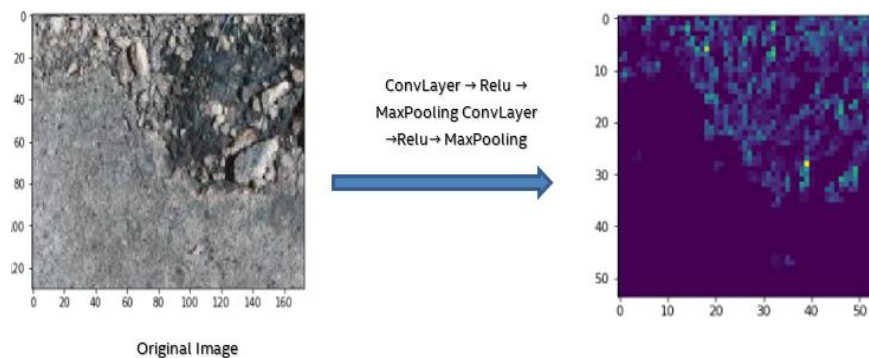


Рисунок 1 - Розпізнавання пошкоджень матеріалу за допомогою CNN

Визначення матеріалів поганої якості у виготовленні апаратного забезпечення – це процес схильний до помилок та вимагає великих затрат часу, що також спричиняє велику кількість помилково позитивних значень (false positives, далі FP), тобто віднесення браку до категорії якісної продукції. Тому у статті [2] пропонують декілька підходів вирішення цієї проблеми.

Перший з них полягає в поєднанні чистого CV для виявлення ділянок зацікавлення (region of interest, далі ROI) із вхідного зображення та чистого DL для визначення дефектів на цих ділянках. В такому порядку дії виконують через те, що експериментальним шляхом було визначено, що продуктивність DL суттєво підвищується за умови фокусування нейронної мережі лише на певних ділянках замість повної площині (рис. 2).

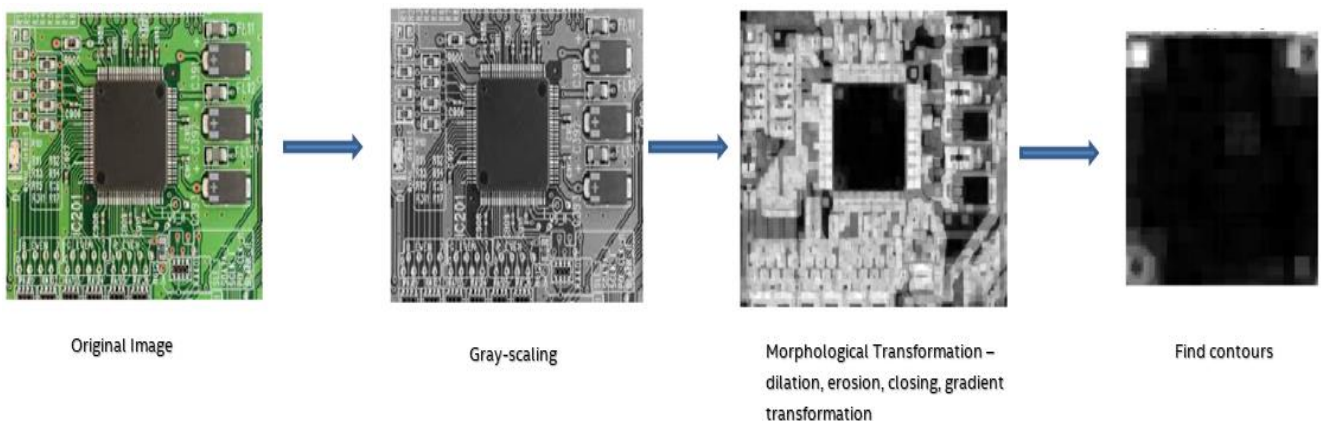


Рисунок 2 - Фокусування нейронної мережі

Надалі виявлення дефектів відбувається за допомогою глибоких нейронних мереж використовуючи перевірені топології Inception Net (Google Net), Res Net, Dense Net (рис. 3).

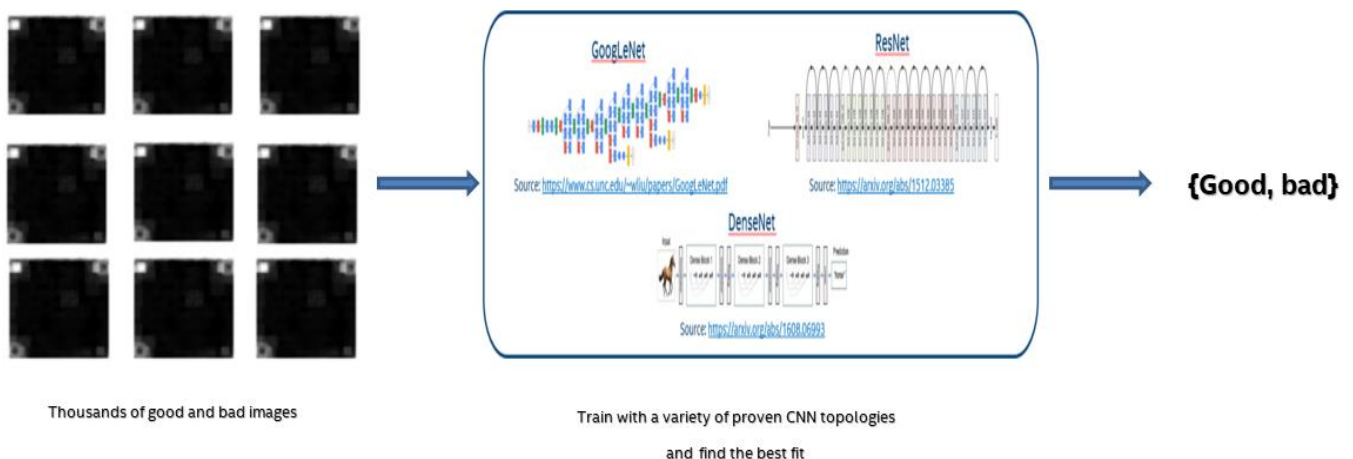


Рисунок 3 - Процес двокрокового розпізнавання дефектів

Другий підхід полягає в критиці до першого. Під час фокусування на ROI необхідно переписувати код незалежно від того чи є зміни щодо типу виробу, мікросхеми, чипу (як в даному прикладі), налаштувань та напрямів камери тощо, тобто масштабування відсутнє. Вирішення: будують закінчену двокрокову архітектуру DL.

На першому кроці замість технік CV використовують DL для передбачення самих ROI. Вручну створюють маркований набір даних за допомогою обмежуючого інструменту, а тоді тренують архітектуру DL для передбачення ROI. Один з мінусів такої техніки є, те що маркований набір даних має бути чітким та охоплювати достатню кількість різних типів продуктів для того, щоб глибока нейронна мережа була здатна добре узагальнювати отримуючи нові зображення (рис. 4).

Оптимальна модель – це завжди компроміс між метриками FPR (false positive rate) та FNR (false negative rate) або Precision та Recall. Для даного випадку головним є успішне виявлення дефектів, а отже модель оптимізують у напрямку нижчого значення FNR (Вищий значення метрики Recall).

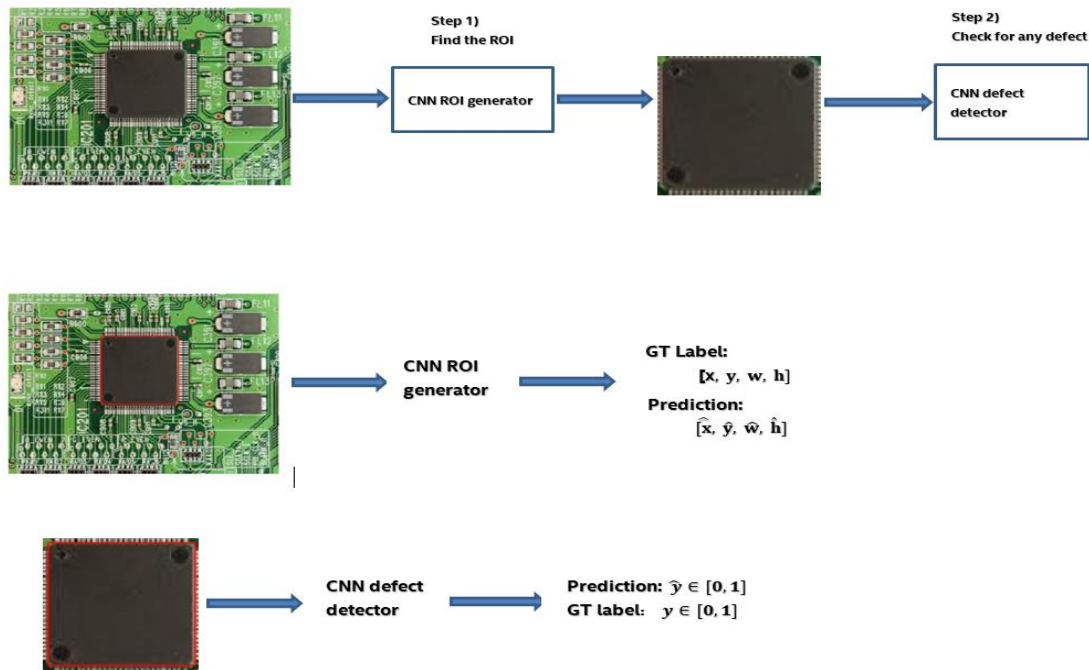


Рисунок 4 - Процес двокрокового розпізнавання дефектів з передбаченням ROI

Окрім перевірки якості у виготовленні матеріальних виробів, ML також широко застосовується для оцінки якості та надійності при розробці програмного забезпечення [6].

Баг означає неочікувану поведінку системи. Така поведінка визначається під час тестування та маркується, як дефект (рис. 5). Виявлення дефектів та виправлення наслідків їх появи – вимагають затрат цінного часу розробки. Досліджено, що лише невелика частина модулів містять більшість недоліків програмного забезпечення [7]. Отже, вчасне виявлення таких дефектів полегшує розподіл ресурсів тестування та дозволяє розробнику покращити архітектуру системи з а допомогою визначення сегментів високого ризику [8].

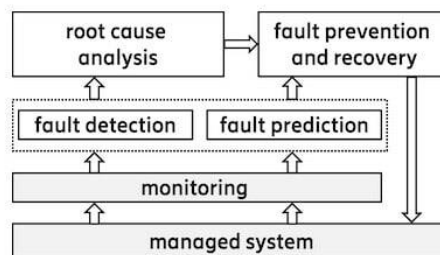


Рисунок 5 - Компоненти системи управління дефектами

Згідно з дослідженнями [6] застосовують різні підходи ML для виявлення багів програмного забезпечення. Для отримання результатів застосовано 3 типи метрик та 5 різних наборів даних. Було доведено ефективність SVM, MLP та технік бегінгу для такого типу задач, що дають в середньому 90% точності при виявленні дефектів. Для вибору найбільш відповідного методу для передбачення багів дослідники радять звертати увагу на різноманітні фактори, такі як: тип набору даних, область проблеми, неоднозначність даних та природу самого проекту.

Також застосовують комплексні підходи ML при тестуванні програмного забезпечення.

Test Scenario Mining. Метод, що дозволяє виконувати тестові завдання, що базуються на тих ділянках, що мають потенційно високу ймовірність помилковості. Це сучасний підхід, що заснований на алгоритмі тестування на основі ризику, який є вирішуючим фактором для виконання тесту. Така техніка підвищує ефективність тестування, а також скорочує затрати, адже дефекти визначаються на ранніх стадіях.

Test Suite Optimizer (Оптимізатор набору тестів). ML використовується також для оптимізації набору тестів. Зростаючий репозиторій тестів з великою кількістю тестової інформації створює надлишковість даних, яку можливо скоротити на 15% за допомогою оптимізатора набору тестів. Це зменшує кількість необхідної роботи за рахунок скорочення регресивного набору для виконання тестових задач у вікнах за фіксований проміжок часу. Структуроване тестування на основі ризику перевіряє додаток, що тестується на всі можливі помилки та ризики.

Analysis of impact (Аналіз впливу). Метод визначає вплив меншої участі експерта, відсутність діаграм UML, встановлення взаємозв'язків між елементами в документі. Машинне навчання пропонує такі переваги як полегшену передачу знань, що дозволена на фазі підтримки, забезпечує полегшений аналіз, ефективне визначення дефектів, що також включає пріоритезацію багів на основі частоти їх появи, серйозності, ризику.

Customer Sentiment Analytics (Аналіз відгуків користувачів). Даний метод використовується для отримання відгуків кінцевого користувача, його бачення та потреб. Такий підхід допомагає знайти точні проблеми, що направляють поведінку користувача. Також це забезпечує навчання в режимі реального часу для отримання постійного зворотного зв'язку, що одночасно покращує менеджмент ризиками.

Висновки та перспективи подальшого дослідження. З використанням DL на основі CV досягають точності діагностування браку у виробництві рівня людського спостерігача в обох підходах CV + DL та DL + DL. Таке рішення є унікальним, адже тут DL використовується не лише для класифікації, а також і для відображення пошкоджених ділянок за допомогою теплових карт. Хоча людський фактор не може бути цілком відкинутий, проте втручання експерта можна суттєво мінімізувати. Також за допомогою ML можливо покращити якість програмного забезпечення, а його впровадження в процес розробки дозволяє розробляти кращі системи на основі відгуків та досвіду користувачів. Підтримка високої якості у створенні продукції – стратегічна ціль для виробників, адже це прямим чином впливає на репутацію та дохід всієї компанії. Інтеграція технологій ML у процес управління якістю може мінімізувати недоліки продукції та скоротити виробничі затрати. Більше того, так як ML – система самонавчання, з часом вона гарантовано покращуватиме власні результати. Подальші дослідження можуть бути спрямовані на застосування Quality 4.0 для розглянутих вище задач.

Список бібліографічного опису

1. Oren Erza, How Machine Learning Slashes Quality Control Costs in Manufacturing [Електронний ресурс]. - Режим доступу: <https://blog.seebo.com/machine-learning-quality-control/>.
2. D. Juran, Quality 4.0: The Future of Quality? [Електронний ресурс]. - Режим доступу: <https://www.juran.com/blog/quality-4-0-the-future-of-quality/>.
3. Graham Immerman, The actual cost of downtime in the manufacturing industry [Електронний ресурс]. - Режим доступу: <https://iiot-world.com/predictive-maintenance/the-actual-cost-of-downtime-in-the-manufacturing-industry/>
4. Partha Deka, Quality inspection in manufacturing using deep learning based computer vision [Електронний ресурс]. - Режим доступу: <https://towardsdatascience.com/quality-inspection-in-manufacturing-using-deep-learning-based-computer-vision-daa3f8f74f45>.
5. Convolutional Neural Network Architecture: Forging Pathways to the Future [Електронний ресурс]. - Режим доступу: <https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-architecture-forging-pathways-future/>
6. S. Aleem, L. Fernando Capretz, A. Faheem, Benchmarking Machine Learning Techniques for Software Defect Detection [Електронний ресурс]. - Режим доступу: https://www.researchgate.net/publication/279252183_Benchmarking_Machine_Learning_Techniques_for_Software_Defect_Detection.
7. N. E. Fenton & N. Ohlsson (2000) "Quantitative analysis of faults and failures in a complex software system", IEEE Transactions on Software Engineering, с. 797- 4.
8. T. Menzies, J. Greenwald & A. Frank (2007) "Data mining static code attributes to learn defect predictors", IEEE Transaction Software Engineering., с. 2-13.

9. Гринюк С.В., Бортник К.Я., Міскевич О.І., Паливода Д.І. Огляд інструментальних засобів для створення ігор під ОС Android. / Комп'ютерно-інтегровані технології: освіта, наука, виробництво. №35., ст. 124-128, 2019р.
10. Міскевич О.І., Сичов Д.І., Христинець А.О. Про модернізацію локально-обчислювальної мережі на ПрАТ "Волиньобленерго" на основі GRE-tunnel з використанням шифрування IPSec. / Комп'ютерно-інтегровані технології: освіта, наука, виробництво. №30-31., ст. 100-103, 2018р.
11. Міскевич О.І., Войтович І.В. Формати зображень та доцільність їх використання в сучасному світі. / Комп'ютерно-інтегровані технології: освіта, наука, виробництво. №38., ст. 85-90, 2018р.
12. Христинець Н.А. Дослідження методу вібраційної сегрегації у формуванні градієнтної структури порошкових матеріалів. // Рудь В.Д., Христинець Н.А. // [Матеріали Міжнародна конференція молодих науковців «Сучасні технології в механіці», – 21-23 квітня 2016 року], м. Хмельницький. /Вісник Хмельницького національного університету, 2016. – Вип.3. – С. 34-41.
13. Христинець Н.А., Рудь В.Д. Стохастичні методи моделювання процесів вібраційного змішування у сипких середовищах. Міжвузівський збірник "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" – Луцьк: Видавництво ЛНТУ. – Вип. 7. – 2011. – 96–98.
14. Христинець Н.А., Рудь В.Д., Колядинський М.І. Модель поведінки часток сипкого середовища під дією вібраційної сегрегації. Міжвузівський збірник "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" – Луцьк: Видавництво ЛНТУ. – Вип. 7. – 2011. – 99–103.

References

1. Oren Ezra, How Machine Learning Slashes Quality Control Costs in Manufacturing [Електронний ресурс]. - Режим доступу: <https://blog.seebo.com/machine-learning-quality-control/>.
2. D. Juran, Quality 4.0: The Future of Quality? [Електронний ресурс]. - Режим доступу: <https://www.juran.com/blog/quality-4-0-the-future-of-quality/>.
3. Graham Immerman, The actual cost of downtime in the manufacturing industry [Електронний ресурс]. - Режим доступу: <https://www.machinemetrics.com/blog/the-real-cost-of-downtime-in-manufacturing>.
4. Partha Deka, Quality inspection in manufacturing using deep learning based computer vision [Електронний ресурс]. - Режим доступу: <https://towardsdatascience.com/quality-inspection-in-manufacturing-using-deep-learning-based-computer-vision-daa3f8f74f45>.
5. Convolutional Neural Network Architecture: Forging Pathways to the Future [Електронний ресурс]. - Режим доступу: <https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-architecture-forging-pathways-future/>
6. International Journal of Software Engineering & Applications (IJSEA), Vol.6, No.3, May 2015
7. S. Aleem, L. Fernando Capretz, A. Faheem, Benchmarking Machine Learning Techniques for Software Defect Detection [Електронний ресурс] - Режим доступу: https://www.researchgate.net/publication/279252183_Benchmarking_Machine_Learning_Techniques_for_Software_Defect_Detection
8. N. E. Fenton & N. Ohlsson (2000) "Quantitative analysis of faults and failures in a complex software system", IEEE Transactions on Software Engineering, pp. 797-814.
9. T. Menzies, J. Greenwald, A. Frank (2007) "Data mining static code attributes to learn defect predictors", IEEE Transaction Software Engineering, pp. 2-13.
10. 6. S.V. Grynyuk, K.Ya. Bortnik, O.I. Miskevych, D.I. Palivoda An overview of tools for creating games on Android. / Computer-integrated technologies: education, science, production. No. 35, Art. 124-128, 2019.
11. 7. Miskevych O.I., Sychev D.I., Khrystynets N.A. About modernization of the local area network on PJSC "Volynoblenergo" based on GRE-tunnel using IPSec encryption. / Computer-integrated technologies: education, science, production. №30-31., Art. 100-103, 2018
12. Miskevych O., Ilya Voytovich. Image formats and the appropriateness of their use in the modern World. / Computer-integrated technologies: education, science, production. №38., Art. 85-90, 2018
13. N. A. Khrystynets, A. A. Sakhnyuk, E. A. Sviridyuk, O. I. Miskevich. Use of bem-blocks when creating a site. / Computer-integrated technologies: education, science, production. №35., Art. 206-210, 2019
14. N. A. Khrystynets, Rud V.D Stochastic methods for modeling vibration mixing processes in bulk media./ Computer-integrated technologies: education, science, production. №7., Art. 96-98, 2011
15. N. A. Khrystynets, Rud V.D, Kolyadinsky M.I. Model of behavior of particles of bulk medium under the action of vibration segregation./ Computer-integrated technologies: education, science, production. №7., Art. 99-103, 2011