Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра математики факультету інформатики

КУРСОВА РОБОТА

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 122 «Комп’ютерні науки та інформаційні технології»

на тему:

“Прогнозування та аналіз часових рядів”

Керівник курсової роботи:
доцент, к.н. Щестюк Н.Ю
Виконала студентка: Женчак А.В

Київ 2020

Зміст

Анотація

В даній роботі розглянуто методи прогнозування часових рядів, що пояснюють поведінку часового ряду, виходячи лише з його значень в попередні моменти часу. Для цього випадку добре підходять моделі ARIMA та нейронні мережі LSTM. Вони добре описують як стаціонарні, так і нестаціонарні часові ряди (більшість часових рядів можуть бути приведені до стаціонарного ряду шляхом виділення тренду, сезонної компоненти, чи взяття різниці).

Мета даного проекту — порівняння методів за допомогою моделей ARIMA та за допомогою нейронних мереж, а саме LSTM. Здійснити аналіз на різних даних, виходячи з їх унікальних форм, щоб перевірити різноманітні зміни в сезоні, підвищення цін та різкі відмінності.

Вступ

На сьогоднішній день виникає потреба в прогнозуванні практично в усіх сферах нашого життя. У нашому житті можуть викликати інтерес завтрашнього курсу валют, прогнозів що до фондового ринку, популярність туристичних послуг, новинок у сфері технологій, прогнозів економічної ситуації в країні і т.д. Все змінюється з часом.

Саме це дало поштовх створенні та розробці великої кількості методів прогнозування даних. Через стрімкий розвиток технологій, пришвидшився розвиток у сфері аналізу великого обсягу даних й почали якісніше створювати різноманітні математичні моделі.

Науковці, що займаються питаннями розвитку економіки, почали працювати над новими теоріями. Адже це допоможе їм відслідкувати напрямок руху економічних змінних та спрогнозувати моменти підйому та спаду.

Найважливіше завдання прогнозування явищ і процесів - виявлення закономірностей і встановлення основних тенденцій розвитку. Для аналізу загальних тенденцій не доцільно розглядати кожен випадок окремо. Чим більша наявність інформаційних даних, тим, за інших рівних умовах, якісніше проявляється закономірність, притаманна досліджуваного явища або процесу. Стійкі пропорції в економічних явищах і процесах проявляються при дії закону великих чисел.

Моделювання та прогнозування дозволяють управляти економічними явищами, процесами та передбачити їх подальший розвиток.

Для моделювання та прогнозування соціально-економічних явищ і процесів вирішальне значення має принцип взаємної їхнього зв'язку та обумовленості. Для того, щоб глибше зрозуміти явище, необхідно вивчити

зовнішні і внутрішні причинні взаємозв'язки, пізнати конкретний стан і умови його виникнення та існування.

Найважливіше завдання прогнозування явищ і процесів - виявлення закономірностей і встановлення основних тенденцій розвитку. Для аналізу загальних тенденцій не доцільно розглядати кожен випадок окремо. Чим більша наявність інформаційних даних, тим, за інших рівних умовах, якісніше проявляється закономірність, притаманна досліджуваного явища або процесу. Стійкі пропорції в економічних явищах і процесах проявляються при дії закону великих чисел.

Моделювання та прогнозування дозволяють управляти масовими економічними явищами, процесами та передбачити їх подальший розвиток.

Для моделювання та прогнозування соціально-економічних явищ і процесів вирішальне значення має принцип взаємної їхнього зв'язку та обумовленості. Для того, щоб глибше зрозуміти явище, необхідно вивчити зовнішні і внутрішні причинні взаємозв'язки, пізнати конкретний стан і умови його виникнення та існування.

Після попереднього вивчення об'єкта переходять до вибору моделі, який можна здійснити як на інтуїтивних, так і на логічних підставах.

Застосування розглянутої в даній роботі методології аналізу і прогнозування на основі часових рядів має досить широке прикладне значення і може використовуватися при вирішенні конкретних завдань дослідження реальних соціально-економічних явищ і процесів як моделювання так і прогнозування.

Ця робота складається з чотирьох розділів, в кожному з яких описані основні прийоми та аспекти виконання практичних завдань, які, зрештою, будуть складені у загальну картину розробки.

Перший розділ містить детальний опис часових рядів та його елементів в процесі аналізу.

Другий розділ складається з описом методів прогнозування та детальної характеристики їхньої дії.

В третьому розділі описана реалізація проекту. Характерні особливості роботи програми будуть показані як блоки коду в додатках з примітками для кращого розуміння його дії. Приклади даних, які використовувалися для дослідження, знаходяться в додатках.

Останній розділ містить результати прогнозування, проведених на трьох наборах даних у реальному часі. Також здійснений їхній аналіз та підведені висновки.

1. Часовий ряд і його основні компоненти

1.1 Часовий ряд

Часовий ряд - це сукупність значень якого-небудь показника за кілька послідовних моментів або періодів часу. Він математично визначається як набір векторів x (t), t = 0,1,2, ... де t — це час, що минув. Змінна x (t) трактується як випадкова величина. Вимірювання, проведені під час події у часовому ряду, розташовані у відповідному хронологічному порядку.

Часовий ряд – одна з реалізацій випадкового процесу, яка реально спостерігається. А випадковим процесом називають набір випадкових величин на якомусь проміжку T, або сукупність реалізацій.

$\xi(t) = \xi(t, \omega): [T \times \Omega] \rightarrow R^1$

Основним завданням статистичного аналізу часових рядів є побудова математичної моделі цього ряду для того, щоб здійснити прогноз на майбутні події.

Часовий ряд, що містить записи однієї змінної, називається універсальним. Але якщо розглядати записи з більш ніж однієї змінної, тоді це називається багатоваріантним (мультиплікативним) часовим рядом.

Часовий ряд може бути безперервним або дискретним. У безперервному часовому ряді спостереження вимірюються в у всіх моментах часу, тоді як дискретний часовий ряд містить спостереження, виміряні в окремі моменти часу. Наприклад, показання температури, течії річки, концентрації хімічного процесу тощо можуть бути записані як безперервний часовий ряд. З іншого боку, населення конкретного міста, виробництво компанії, обмінний курс між двома різними валютами може представляти дискретний часовий ряд. Зазвичай у дискретному часовому ряді послідовні спостереження реєструються з однаково розташованими інтервалами часу, такими як погодинний, щоденний, щотижневий, щомісячний або щорічний поділ часу. Як зазначалося в вище, змінна, яка

спостерігається в дискретному часовому ряді, передбачається вимірювати як суцільну змінну, використовуючи реальну шкалу чисел. Крім того, безперервний часовий ряд може бути легко перетворений у дискретний, об'єднуючи дані разом протягом визначеного часового інтервалу.

1.2 Елементи часового ряду

На часовий ряд зазвичай впливають чотири основні компоненти, які можна відокремити від спостережуваних даних. В залежності від різного поєднання в досліджуваному процесі або явищі цих чинників, залежність рівнів ряду може приймати різні форми. До таких компонентів відносяться: тренд, цикл, сезонність та нерегулярні компоненти. Короткий опис цих чотирьох компонентів наведено далі.

Загальна тренд тимчасового ряду до збільшення, зменшення або застою протягом тривалого періоду називається Secular Trend або просто тенденцією (Simply Trend). Таким чином, можна сказати, що тенденція - це довгостроковий рух у часовому ряді. Очевидно, що безліч факторів, які впливають на динаміку досліджуваного показника, взяті окремо, можуть надавати різноспрямований вплив на досліджуваний показник. Однак в сукупності вони формують його зростаючу або спадаючу тенденцію. Наприклад, часові ряди, в яких спостерігається зростання кількості населення, кількості будинків у місті тощо, демонструють тенденцію до зростання, тоді як тенденцію до зменшення можна спостерігати в серіях, що стосуються показників смертності, епідемій тощо.

Сезонні зміни часових рядів коливаються протягом року або сезону. Важливими факторами, що спричиняють сезонні зміни, є: клімат та погодні умови, звичаї, календарні святкування, відпустки тощо. Наприклад, збільшення продажів морозива влітку, а продаж вовняних тканин зростає взимку. Сезонні зміни є важливим фактором для бізнесменів та виробників для створення належних планів на майбутнє.

Циклічність часового ряду описує середньострокові зміни ряду, викликані обставинами, які повторюються за циклами. Тривалість циклу може подовжитися на більш тривалий проміжок часу, зазвичай два або більше років. Циклічні коливання можуть бути здатними нести сезонний

характер, оскільки діяльність ряду деяких галузей економіки та сільського господарства залежить від пори року. При наявності великих масивів даних тривалих проміжків часу можна знайти циклічні коливання, пов'язані з загальною динамікою часового ряду. Більшість економічних та фінансових часових рядів демонструють певну циклічну варіативність. Наприклад, бізнес-цикл складається з чотирьох фаз, а саме. і) процвітання, іі) спад, ііі) депресія та іv) одужання.

Схематично типовий бізнес цикл зображений нижче:

Рисунок 1.2 Чотирьох фазовий бізнес цикл

Нерегулярні або випадкові коливання в часових рядах зумовлені непередбачуваними впливами, які не є регулярними, а також не повторюються за певною схемою. Ці зміни обумовлені випадками, такими як війна, страйк, землетрус, повені, революція тощо. Не існує визначеної статистичної техніки для вимірювання випадкових коливань у часовому ряду. Деякі тимчасові ряди не містять тенденції і циклічної компоненти, а кожен наступний їх рівень утворюється як сума середнього рівня ряду і деякої (позитивної або негативної) випадкової компоненти.

Зважаючи на ефекти цих чотирьох компонентів, два різних типи моделей зазвичай використовуються для часового ряду, а саме : мультиплікативні (Multiplicative Model) та додаткові (Additive Model) моделі.

Модель, в якій тимчасовий ряд представлений як сума перерахованих компонент, називається адитивної моделлю часового ряду.

Модель, в якій тимчасової ряд представлений як добуток перерахованих компонент, називається мультиплікативной моделлю часового ряду.

Multiplicative Model: $Y(t) = T(t) \times S(t) \times C(t) \times I(t)$.

Additive Model: $Y(t) = T(t) + S(t) + C(t) + I(t)$.

Тут Y(t) — спостереження, T(t) — тренд S(t), C(t) — циклічність, I(t) — нерегулярні зміни в часі. Мультиплікативна модель заснована на припущенні, що чотири компоненти часового ряду не обов'язково є незалежними, і вони можуть впливати один на одного; тоді як в адитивній моделі передбачається, що чотири компоненти не залежать одна від одної.

Основне завдання статистичного дослідження окремого часового ряду - виявити і надати кількісне вираження кожної з перерахованих вище компонент з тим, щоб використовувати отриману інформацію для прогнозування майбутніх значень ряду.

1.3 Числові характеристики часового ряду

Для вивчення часових рядів зазвичай використовують ймовірнісно-статистичні моделі. Тоді часовий ряд розглядають як це реалізація деякого випадкового процесу з наступними основними характеристиками:

1. $\mu(t) = E(\xi(t))$ − тренд (математичне сподівання),
2. $D(t) = Var(t) = E(\xi - E_{\xi t})^2$ − дисперсія,
3. $\sigma(t) = \sqrt{D(t)}$ − середнє квадратичне відхилення,
4. $R(t_1, t_2) = cov(\xi(t_1), \xi(t_2)) = E(\xi(t_1) - \mu(t))(\xi(t_2) - \mu(t))$ − автоковаріація,
5. $\rho(t_1, t_2) = \frac{cov(\xi(t_1),\xi(t_2))}{\sigma(t_1)\sigma(t_2)}$ − кореляція.

Можна умовно поділити усі випадкові процеси на два види: стаціонарні і нестаціонарні. Стаціонарним часовим рядом називають випадкові процеси y_t, статистичні характеристики яких не змінюються в часі t, тобто інваріантні щодо часових зрушень $t \rightarrow t+T$, $y_t \rightarrow y_{t+1}$ при будь-якому фіксованому Т. Обов'язковою умовою стаціонарності у широкому значені є те, щоб математичне сподівання та автоковаріація були константами. Умовою стаціонарності у вузькому значенні є інваріантність n-вимірної щільності ймовірності відносно часового зсуву τ.

Процес буде вважатись стаціонарним, якщо не виконуються такі умови:

- математичне сподівання не залежить від часу: $E[y_t]=E[y_{t-s}] = \mu = const$;
- дисперсія для всього часового інтервалу, на якому розглядається процес не змінюється : $E\{[y_t - \mu]^2\} = E\{[y_{t-s} - \mu]^2\} = \sigma^2_y = const$;
- автоковаріація для всього часового інтервалу, на якому розглядається процес не змінюється і залежить тільки від вибраних моментів часу t і t-s:

  $E\{[y_t - \mu][y_{t-s} - \mu]\} = E\{[y_{t-j} - \mu][y_{t-j-s} - \mu]\} = \gamma_s = const$

  або $cov[y_t, y_{t-s}] = cov[y_{t-j}, y_{t-j-s}] = \gamma_s = const$.

Процес буде вважатись нестаціонарним, якщо не виконується хоча б одна з умов наведена вище.

Автокореляційна функція — це кореляція функції з самою собою, але зсунутою на деякий проміжок часу, який називається лагом. Основні властивості автокореляційної функції:

1. $R(0) \geq 0$,

2. $R(n) = R(-n)$,

3. $R(0) \geq |R(n)|$.

Відповідником АКФ є вибіркова часткова автокореляційна функція (ЧАКФ), але після вилучення впливу проміжних лагів. Часткова автокореляційна функції(ЧАКФ) є поглибленням поняття звичайної автокореляційної функції. Часткова автокореляція дає більш «чисту» картину періодичних залежностей.

ЧАКФ 1-го порядку:

$y_t = \varphi_{11} y_{t-1} + \varepsilon_t$

ЧАКФ 2-го порядку:

$y_t = \varphi_{21} y_{t-1} + \varphi_{22} y_{t-2} + \varepsilon_t$

1.4 Приклади часових рядів

Спостереження за тимчасовими рядами часто зустрічаються у багатьох сферах, таких як бізнес, економіка, промисловість, інженерія та наука тощо. Залежно від характеру аналізу та практичної потреби, можуть бути різного роду часові ряди. Для візуалізації основної схеми даних, як правило, часовий ряд представлений графіком, де спостереження будуються відповідно до відповідного часу. Нижче зображено два сюжети часових рядів:

Рисунок 1.3.1: Щотижневі дані курсів валют GBP / USD (1980-1993)

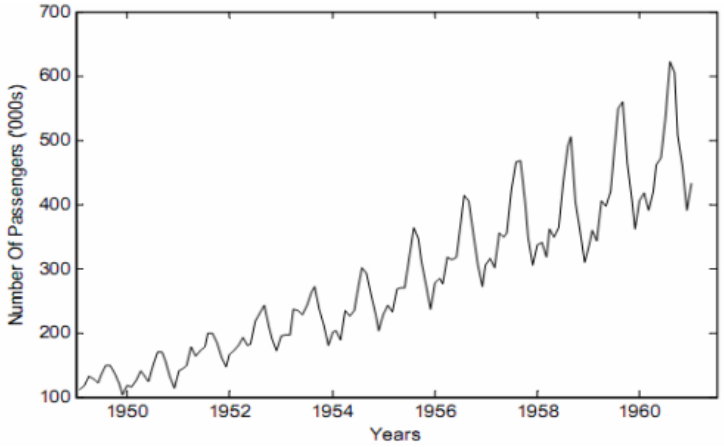


Рисунок 1.2.2 : Щомісячні міжнародні пасажирські рейси авіакомпаній (січень 1949 - грудень 1960)

Перший часовий ряд представляє тижневий курс між британським фунтом і доларом США з 1980 по 1933 рр.

Другий - сезонний часовий ряд, який показує кількість пасажирів міжнародних авіакомпаній (у тисячах) за період з січня 1949 по грудень 1960 щомісяця.

2. Методи аналізу часових рядів

На практиці модель підходить до заданого часового ряду і її відповідні параметри оцінюються за допомогою відомих значень ряду. Процедуру пристосування часових рядів до належної моделі називають аналізом часових рядів. Він містить методи, які намагаються зрозуміти природу поведінки ряду і часто корисні для майбутнього прогнозування та моделювання.

Під час прогнозування часових рядів попередні спостереження збираються та аналізуються для розробки відповідної математичної моделі, яка фіксує основні процеси генерування даних для ряду. Далі майбутні події прогнозуються за допомогою моделі.

Цей підхід є особливо корисним, коли недостатньо знань про статистичну схему, за якою слідують послідовні спостереження, або коли бракує задовільної пояснювальної моделі. Прогнозування часових рядів має важливе застосування в різних областях. Часто цінні стратегічні рішення та запобіжні заходи приймаються на основі результатів прогнозу. Таким чином, зробити хороший прогноз, тобто пристосувати відповідну модель до часового ряду, є досить важливим. За останні кілька десятиліть дослідники доклали багатьох зусиль для розробки та вдосконалення відповідних моделей прогнозування часових рядів.

Часовий ряд несе недетермінований характер, тобто ми не можемо з впевненістю передбачити, що відбуватиметься в майбутньому. Зазвичай часовий ряд {x (t), t = 0,1, 2, ...} передбачає наявність певної моделі ймовірності, яка описує спільний розподіл випадкової величини x_t. Математичний вираз, що описує структуру ймовірності часового ряду називається стохастичним процесом. Таким чином, послідовність спостережень серії насправді є зразковою реалізацією стохастичного процесу, який її спричинив.

Звичайним припущенням є те, що змінні часових рядів x_t є незалежними та однаково розподіленими (independent and identically distributed , i.i.d) після нормального розподілу. Однак, як згадувалося раніше, цікавим моментом є те, що часові ряди насправді не є саме i.i.d; вони дотримуються більш-менш деякої регулярної структури в довгостроковій перспективі. Наприклад, якщо сьогодні температура в певному місті надзвичайно висока, то можна обґрунтовано припустити, що завтрашня температура також буде високою. З цієї причини прогнозування часових рядів за допомогою відповідної методики дає результат, близький до фактичного значення.

2.1. Авторегресійні моделі

Існує декілька найбільш широко використовуваних основних моделей стаціонарних процесів. До них відносять:

- авторегресія (AR)
- зсувне середнє (MA)
- ARMA
- ARIMA

Авторегресійна модель (Autoregressive model, AR) — модель часових рядів, в якій значення часового ряду в даний момент лінійно залежать від попередніх значень цього ж ряду. Авторегрессіонний процес порядку p визначається наступним чином:

$$X_t = c + \sum_{i=1}^{p} a_i X_{t-i} + \varepsilon_t,$$

Рисунок 2.1: Формула AR

де a_1 , ... , a_p – параметри моделі (коефіцієнти авторегресії); c – константа; ε_t – білий шум.

Зсувне середнє (Moving Average, MA) — функція, значення якої в кожній точці визначене, як середнє значення початкової функції за попередній період.

$$X_t = c + \sum_{j=0}^{q} b_j \varepsilon_{t-j},$$

Рисунок 2.2: Формула MA

де b_0, ... , b_q – параметри моделі; c – константа; ε_t – білий шум.

Модель авторегресії ковзаючого середнього, Autoregressive integrated moving average (ARMA) — математична модель аналізу та прогнозування стаціонарних часових рядів, є узагальненням моделі авторегресії та моделі ковзаючого середнього.

Моделлю ARMA(p, q) , де p і q – цілі числа, що задають порядок моделі, називається наступний процес генерації тимчасового ряду X_t :

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} a_i X_{t-i} + \sum_{j=0}^{q} b_j \varepsilon_{t-j} ,$$

Рисунок 2.3: Формула ARMA

де a_1, ... , a_p – коефіцієнти авторегресії; b_0, ... , b_q – коефіцієнти зсувного середнього; c – константа; ε_t – білий шум.

Лінійні параметричні моделі дістали загальну назву авторегресійні інтегровані моделі зсувного середнього (ARIMA). Вони ґрунтуються на припущенні лінійності процесу породження даних і описують стаціонарний процес, який має три ознаки:

- p — порядок авторегресії AR;
- d — порядок послідовних різниць рівнів часових рядів, що забезпечує стаціонарність ряду;
- q — порядок зсувного середнього MA.

ARIMA є розширенням моделей ARMA для нестаціонарних часових рядів, які можна зробити стаціонарними взяттям різниць деякого порядку від вихідного часового ряду (так звані інтегровані або різницево-стаціонарні тимчасові ряди). Має наступний вигляд:

$$\Delta^d X_t = c + \varepsilon_t + \sum_{i=1}^{p} a_i \Delta^d X_{t-i} + \sum_{j=0}^{q} b_j \varepsilon_{t-j} ,$$

Рисунок 2.4: Формула ARIMA

Δ^d – оператор різниці часового ряду порядку d (послідовне взяття d раз d різниць першого порядку – спочатку від тимчасового ряду, потім від отриманих різниць першого порядку, потім від другого порядку і т.д.)

При p = 1, 2, ..., n d = 0 та q = 0, модель ARIMA перетворюється на процес AR. ARIMA (p, 0, 0) = AR(p)

Якщо p = 0, d = 0 та q = 1, 2,..., n, то маємо модель MA. ARIMA (0, 0, q) = MA(q).

2.2. Метод LSTM

Нейромережеві методи — це методи або математичні, що ґрунтуються на штучних нейронних мережах. Нейромережеві моделі можна розглядати як прості математичні моделі, що визначають функцію $f : X \rightarrow Y$, або розподіл над X , або над X та Y . Іноді моделі тісно пов'язують з певним правилом навчання.

Штучна нейронна мережа являє собою шарувату структуру з пов'язаних нейронів. Нейронна мережа - це послідовність нейронів, з'єднаних між собою синапсами. Структура цієї мережі прийшла в світ програмування завдяки біологічним процесам. Це дозволяє машині аналізувати, запам'ятовувати різну інформацію і також відтворювати її зі своєї пам’яті. Це не один алгоритм, а комбінація різних алгоритмів, яка дозволяє нам виконувати складні операції з даними.

Рекурентні нейронні мережі (Neural Network, RNN) – це мережі, у яких з’єднання між вузлами утворють орієнтований цикл. Це створює внутрішній стан мережі, що дозволяє їй проявляти динамічну поведінку в часі. У RNN існують повторювані модулі «tanh» шарів, які дозволяють їм зберігати інформацію. На відміну від нейронних мереж прямого поширення, RNN можуть використовувати свою внутрішню пам'ять для обробки довільних послідовностей входів. Це робить їх застосовними до таких задач, як розпізнавання несегментованого неперервного рукописного тексту та розпізнавання мовлення.

За останні декілька років RNN неймовірним успіхом застосували до цілого ряду завдань:розпізнавання мови, мовне моделювання, переклад, розпізнавання зображень. Чимала роль у цих успіхах належить LSTM — незвичайна модифікація рекурентної нейронної мережі,яка на багатьох задачах значно перевищує стандартну версію. Майже всі вражаючі результати RNN досягнуті саме за допомогою LSTM.

Довга короткострокова пам'ять (Long Short-Term Memory, LSTM) — це особливий вид RNN, що складається з набору комірок з функціями для запам'ятовування послідовності даних. Комірка захоплює і зберігає потоки даних. Далі ці комірки з'єднують один модуль з минулого разом з іншим модулем теперішнього, щоб передавати інформацію з декількох моментів минулого часу до теперішнього. Завдяки використанню воріт у кожній комірці дані в кожній комірці можуть бути розміщені, відфільтровані або додані для наступних комірок.

Ворота, засновані на сигмоїдальному шарі нейронної мережі, дозволяють коміркам необов'язково пропускати дані або розміщуватись. Кожен сигмоподібний шар дає числа в діапазоні від нуля до одиниці, зображуючи кількість кожного сегмента даних, який слід пропустити в кожну клітинку. Точніше, оцінка нульового значення означає, що "нехай нічого не пройде"; тоді як; оцінка 1 вказує на те, що "нехай все пройде". В кожній LSTM задіяні три типи воріт з метою контролю за станом кожної комірки:

- Forget Gate видає число від 0 до 1, де 1 показує "повністю зберегти це"; тоді як 0 означає «повністю ігнорувати це».
- Memory Gate вибирає, які нові дані потрібно зберігати в комірці через сигмоподібний шар, за яким йде шар tanh. Початковий сигмоїдний шар, який називається "вхідний дверний шар", вибирає значення, які будуть змінені. Далі, шар tanh робить вектор нових кандидатських значень, які можна додати.
- Output Gate визначають, який буде вихід з кожної комірки. Вихідне значення буде базуватися на стані комірки разом з відфільтрованими та нещодавно доданими даними.

Ще одна відмінність LSTM і RNN полягає в тому що, повторюваний модуль в стандартній RNN складається з одного шару, а в LSTM — чотирьох.

Штучний нейрон являє собою одиницю обробки інформації у нейронній мережі. Поведінка нейрона будується наступним чином: нехай є m + 1 входів, значення яких дорівнюють x_0 , x_1 , x_2 , ..., x_m, а значення їх ваг рівні, w_0 ,w_1 ,...,w_m, при цьому перший вхідний елемент, як правило, являє собою фіксоване значення зміщення $x_0 = 1$.

LSTM розроблені спеціально, щоб уникнути проблеми довготривалої залежності. Запам'ятовування інформації на довгі періоди часу - це їх звичайнаповедінка.

LSTM визначає, які елементи інформації необхідно забути, які додати, а на яких необхідно сфокусувати увагу. Такий підхід дозволяє відстежувати інформаціюпротягом більш тривалих періодівчасу.

2.3 Виміри ефективності прогнозування

У попередніх розділах ми вивчали різні корисні та популярні методи прогнозування часових рядів. Наступне важливе питання - реалізація впровадження, тобто застосування цих методів для генерування прогнозів. При застосуванні певної моделі до деякого реального або модельованого часового ряду спочатку необроблені дані поділяються на дві частини, а саме. навчальний набір і тестовий набір.

Спостереження в навчальному наборі використовуються для побудови потрібної моделі. Часто невелика частина навчального набору зберігається з метою перевірки і називається набором валідації (Validation Set). Іноді попередня обробка проводиться шляхом нормалізації даних або прийняття логарифмічних чи інших перетворень. Однією з таких відомих методик є трансформація Box-Cox. Після побудови моделі вона використовується для генерування прогнозів. Спостереження набору тестів зберігаються для перевірки наскільки точно виконана відповідна модель при прогнозуванні цих значень. При необхідності на прогнозовані значення застосовується обернене перетворення для їх перетворення в початковий вид. Для того, щоб оцінити точність прогнозування конкретної моделі або оцінити та порівняти різні моделі, враховується їх відносна ефективність на тестовому наборі даних.

У зв'язку з принциповою важливістю прогнозування часових рядів у багатьох практичних ситуаціях слід уважно ставитися до вибору конкретної моделі. З цієї причини пропонуються різні заходи щодо ефективності оцінки точності прогнозу та порівняння різних моделей. Вони також відомі як показники ефективності. Кожен з цих заходів є функцією фактичних та прогнозованих значень часового ряду.

Значення «втрат», як правило, повідомляються алгоритмами глибинного навчання (deep learning). Технічна втрата - це певна кара за

поганий прогноз. Більш конкретно, значення втрат буде нульовим, якщо прогноз моделі ідеальний. Отже, метою є мінімізація значень втрат за рахунок отримання набору ваг і ухилів, що мінімізує втрати. Крім втрат, які використовуються алгоритмами глибинного навчання, дослідники часто використовують середнє квадратичне відхилення (RMSE) для оцінки показників прогнозування. RMSE вимірює різниці між фактичними та прогнозованими значеннями. Формула для обчислення RMSE така:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - y_i'\right)^2}$$

Де N - загальна кількість спостережень, y_i - фактичне значення; тоді як $\hat{y}_i$ - це прогнозоване значення.

Основна перевага використання RMSE полягає в тому, що він не допускає великих помилок. Він також вимірює бали в тих же одиницях, що і прогнозні значення.

3. Реалізація методів прогнозування

Цей проект написаний на мові Python з метою порівняти ARIMA та LSTM моделі з різними часовими рядами. Він складається з основних методів, які могли б швидко перевірити, інтегрувати та оптимізувати моделі ARIMA та LSTM для заданого часового ряду. Ці моделі використовувались для прогнозування різних даних:

- Alphabet Inc. Yahoo Finance (GOOGL)
- Євробачення (Google Trends)
- Вірус Зіка (Google Trends)
- Lionel Messi (Google Trends)
- Game of Thrones (Google Trends)
- Iphone (Google Trends)

Я вибрала ці тенденції, виходячи з їх унікальних форм, щоб перевірити наявність тренду і його характер, наявність сезонних і циклічних компонент. Кожен частина містить графік вихідних даних з подальшими результатами тестування поза вибіркою.

Далі я включила результати, які використовували фільтрацію гауса для згладжування вихідного набору даних з подальшим моделюванням згладженого набору даних. Згладжені результати моделювання порівнювались з вихідними даними ряду, щоб визначити, чи фільтрування покращило продуктивність моделей.

Для даної роботи були використані бібліотеки pandas і numpy для даних та для їхньої підготовки перед застосуванням моделей; statsmodel і keras для моделювання ARIMA і LSTM

3.1 ARIMA

Функція arima_model створює прогноз прогону моделі ARIMA для заданого часового ряду. Це дозволяє користувачеві вибирати значення p, q і d, а також вказувати, чи слід реєструвати перетворення. Різні етапи функції включають:

- журнал перетворення даних
- поділ на тренувальні та тестові набори даних
- створення моделі ARIMA для тренувального набору даних
- створює прогноз, випробовуючи перше значення в тестовому наборі та додаванням цього значення до навчального набору та з подальшим тестуванням наступного значення у вибірці.
- обернене перетворення даних
- створення графіків та генерування показників помилок (RMSE).

3.2 LSTM

LSTM складається з трьох різних функцій для створення набору даних:

- create_dataset
- inverse_transforms
- lstm_model

За допомогою цих функцій можна моделювати, тренувати та тестувати нейронну мережу LSTM, а потім інвертувати будь-які перетворення даних, щоб можна було побудувати графіки та проаналізувати помилки в початковому вигляді.

Створення вибірки даних(create_dataset)

Спочатку створюється вибірка даних (dataset), необхідниа для тренування та тестування нейронних мереж LSTM. Вона приймає на вхід часовий ряд, кількість попередніх періодів, на основі яких користувач хотів би створити модель, здійснює поділ на тренувальні та тестові частини. Пізніше дані розміщаються в проміжку між 0 і 1 для введення в LSTM модель.

Необхідно навчати t-n часові періоди. Спочатку потрібно навчати дані для поточного місяця (t-0), вхідними даними будуть дані з попереднього місяця (t-1), тоді як цільовим буде фактичне значення для цього поточного місяця (якщо вважати, що це відомо). Збільшення look_back надасть додаткові попередні місяці для включення, наприклад, два місяці тому (t-2) і три місяці тому (t-3).

Зворотнє перетворення (inverse_transforms)

Ця функція зворотного перетворення просто скасовує будь-які перетворення, виконані під час генерації набору даних. Інвертування перетворень дозволяє прогнозувати моделі на основі тієї ж шкали, що і

вихідний набір даних для більш інтуїтивної інтерпретації результатів. Як функція створення вибірки, так і функція зворотного перетворення автоматично викликаються в межах функції LSTM.

LSTM моделювання(lstm_model)

Для виклику моделі LSTM потрібні:

- набір даних часового ряду,
- кількість бажаних періодів огляду,
- розділення на тренувальну та тестову вибірку,
- чи слід запам'ятовувати перетворення чи різницю даних,
- параметри для тренування, такі як кількість вузлів та епох.

У межах функції вона створює набори даних для тренування і тестує датасети, а потім тренує модель LSTM з подальшим прогнозуванням даних. Прогнози моделі, а також фактичні цільові значення потім обернено перетворюються за допомогою функції inverse_transforms, і графіки генеруються разом із показниками помилок.

Гаусова фільткація

Гауссова фільтрація приводить до кращих прогнозів даних про вибірку. Функція gauss_compare дозволяє порівняти відфільтровані дані прогнозу разом з початковими даними за допомогою візуалізації та RMSE-звітування.

4. Аналіз результатів

Я порівнювала оптимізовану моделі ARIMA з моделями LSTM з нефільтрованими даними для кожної, а також з фільтрацією Гаусса.

Оптимізація була здебільшого емпіричною, де певна допомога йшла з автокореляцією та часткової автокореляцією для параметрів ARIMA. Загалом, я оптимізувала значення перетворення p, d, q та log для моделей ARIMA й також log / диференціювання та епохи для моделей LSTM.

Для кожного прикладу графіки показують прогнози з фактичними значеннями тестувальної вибірки, поряд із значеннями RMSE. Для всіх сюжетів фактичний (або відфільтрований) часовий ряд простежується у синім кольором, тоді як зроблені прогнози простежуються у червоному. Я намагалася підібрати різноманітні дані, які включали тренд, сезонність, циклічність, різкий спад, тощо.

Eurovision

Пісенний конкурс Євробачення проводиться щороку на початку травня. Дані взято з ресурсу Google Trends в період з 2004 року до березня 2020. Періодична різка тенденція, пов’язана з датою фіналу, можна побачити у часовій серії нижче. Я вибрала цей часовий ряд для прогнозування через його різкі стрибки, але з циклічною повторюваністю. Методою дослідження було з’ясувати чи дано достатньо подібних даних для навчання моделі і чи зможуть моделі точно дати прогноз.

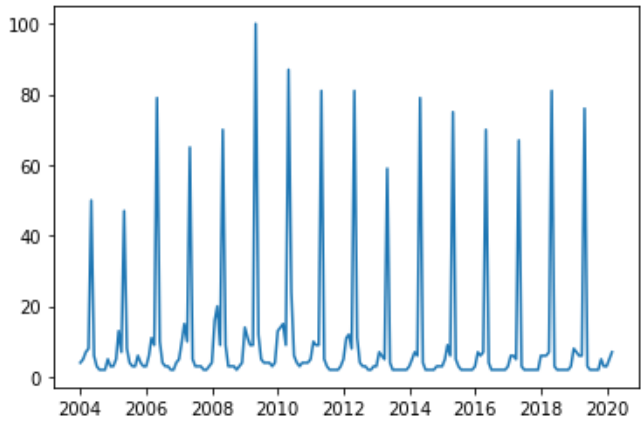


Рисунок 4.1:Часовий ряд Євробачення

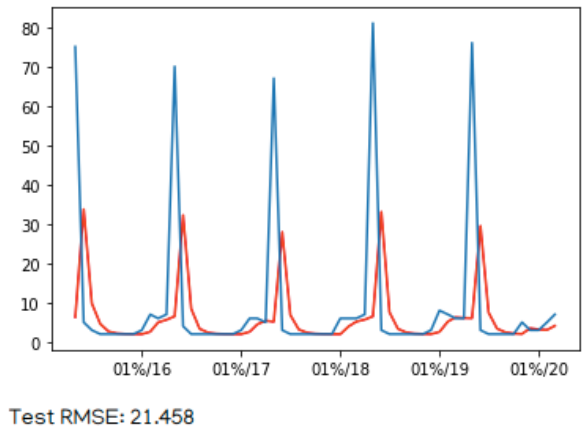
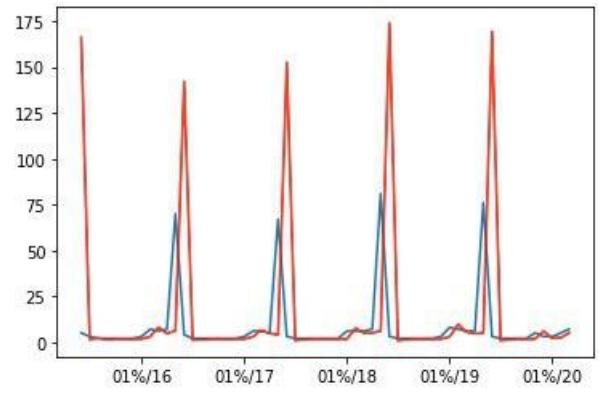


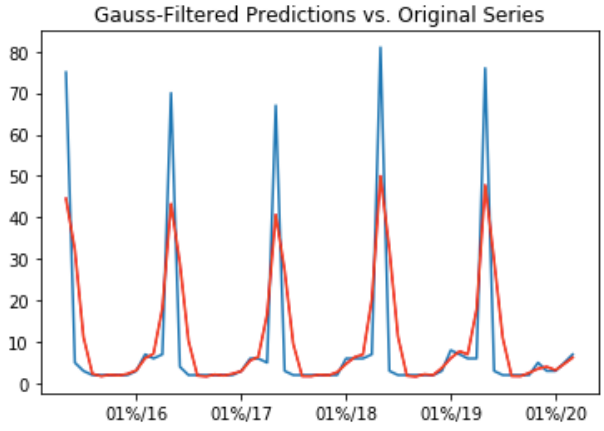
Рисунок 4.2: ARIMA (0,1,1) Євробачення

Модель LSTM вимагала мало епох, щоб навчитися працювати точно, що може бути пов'язано з повторюваним характером серії. Однак у моделі виникли труднощі з визначенням вершини та спадом тенденції.



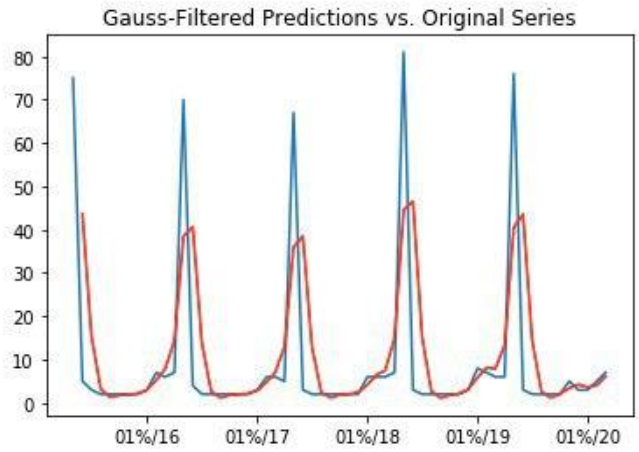
Train RMSE: 44.692
Test RMSE: 49.669

Рисунок 4.3: LSTM Євробачення



Test RMSE: 12.044

Рисунок 4.4: ARIMA Gaussian Filter Євробачення



Test RMSE: 15.088

Рисунок 4.5: LSTM Gausian Filter Євробачення

Цей набір даних отримав найбільші значення RMSE з усіх модельованих часових серій, тобто, незважаючи на повторювальність стрибків та гострі градієнти, все ще були складні для моделювання. ARIMA працювала краще ніж модель LSTM, завдяки фільтруванню Гаусса покращився коефіцієнт помилок приблизно на 50%. Модель ARIMA також мала певні труднощі з конвергенцією для більш високих значень p і q через різкі градієнти на нефільтрованих даних. Короткий огляд RMSE:

ARIMA RMSE (no filter): 21.458

LSTM RMSE (no filter): 49.669

ARIMA RMSE (filtered): 12.044

LSTM RMSE (filtered): 15.088

Вірус Зіка

Вірус Зіка — це такий вірус, який передається через укуси комарів. Дані взято з Google Trends від 2004 року до 2019. Я вибрала цей тренд , щоб перевірити, як алгоритми реагуватимуть на навчальну вибірку з майже відсутньою корисною інформацією (постійні 0), з подальшим різким зростанням інтересу до пошуку.

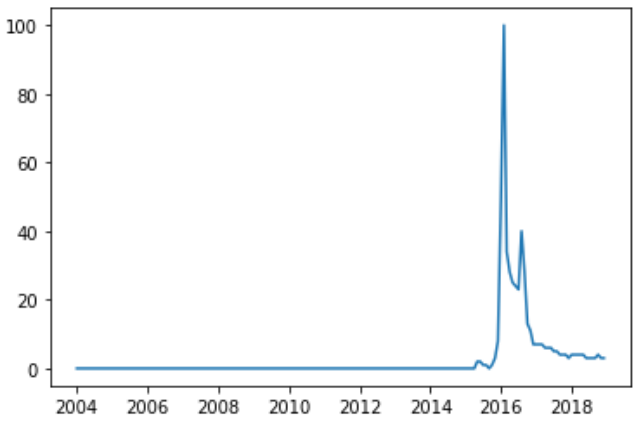
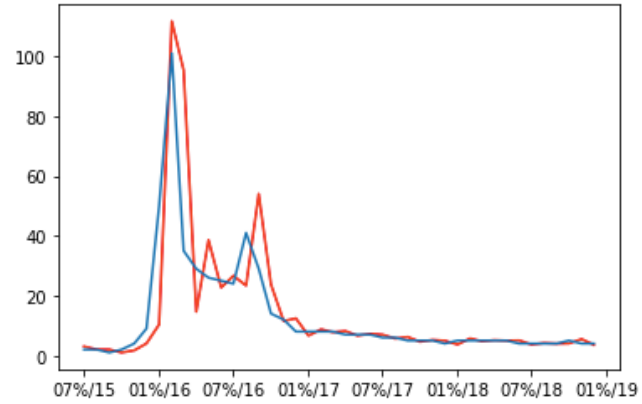
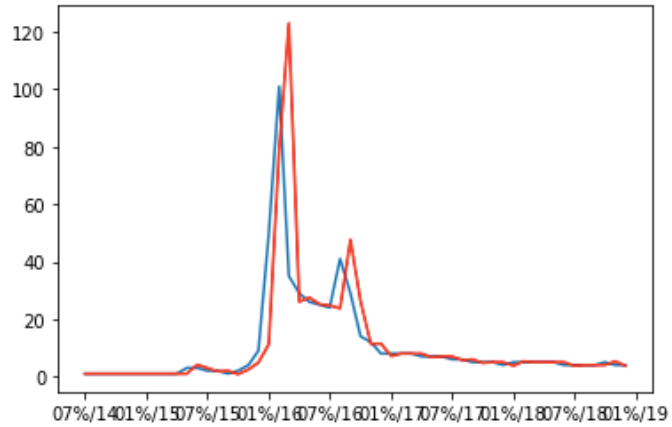


Рисунок 4.6: Вірус Зіка



Test RMSE: 12.727

Рисунок 4.7: ARIMA Вірус Зіка



Train RMSE: 0.000
Test RMSE: 13.996

Рисунок 4.8: LSTM Вірус Зіка

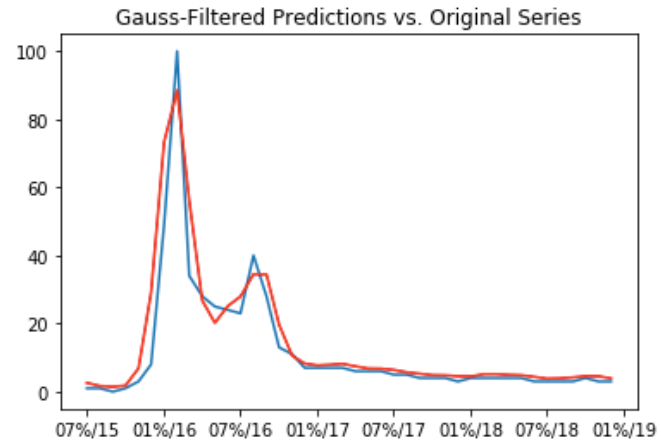


Рисунок 4.9: ARIMA Gaussian Filter Вірус Зіка

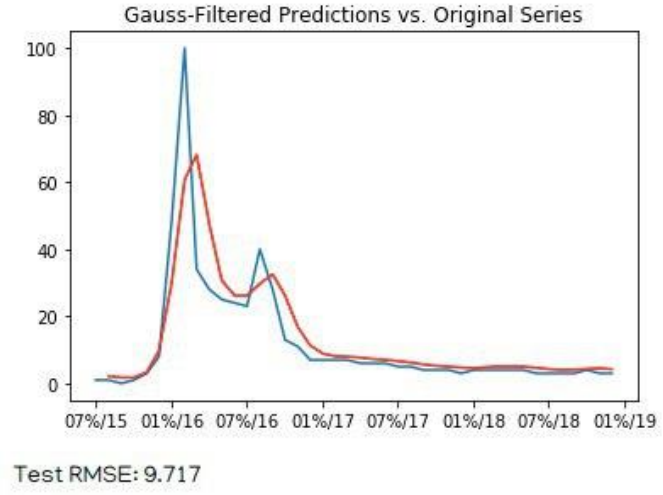


Рисунок 4.10: LSTM Gaussian Filter Вірус Зіка

Модель LSTM була дещо гіршою щодо RMSE, але мала труднощі з різкістю стрибів. Змінивши розмір даних для навчання до 0,77 для того, щоб зафіксувати деякий рух у даних, тобто в іншому випадку я отримала б сингулярну помилку матриці з усіх значень які були б стабільними, це призводить до знаходження марних коефіцієнтів тренувань.

ARIMA RMSE (no filter): 12.727

LSTM RMSE (no filter): 13.996

ARIMA RMSE (filtered): 6.677

LSTM RMSE (filtered): 9.717

Alphabet Inc. (Google)

Alphabet Inc. надає рекламні послуги в інтернеті в США, Європі, на Сході, в Африці, Азіатсько-Тихоокеанському регіоні, Канаді та Латинській Америці. Вона пропонує послуги з ефективності та реклами бренду. Компанія працює через сегменти Google. Сегмент Google пропонує такі продукти, як реклама, Android, Chrome, Google Cloud, Google Maps, Google Play, апаратне забезпечення, пошук та YouTube, а також технічну інфраструктуру. Вона також пропонує цифровий контент, хмарні сервіси, апаратні пристрої та інші різні продукти та послуги.

Нижче наведено дані про ціни GOOGL з 2004 рік по березень 2020 року (отримані за допомогою Yahoo Finance). Ця тенденція показує збільшення значень з часом, а також деякі стрімкі підйоми та спуски.

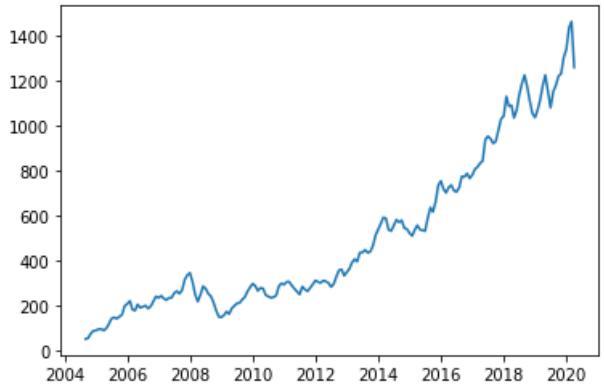
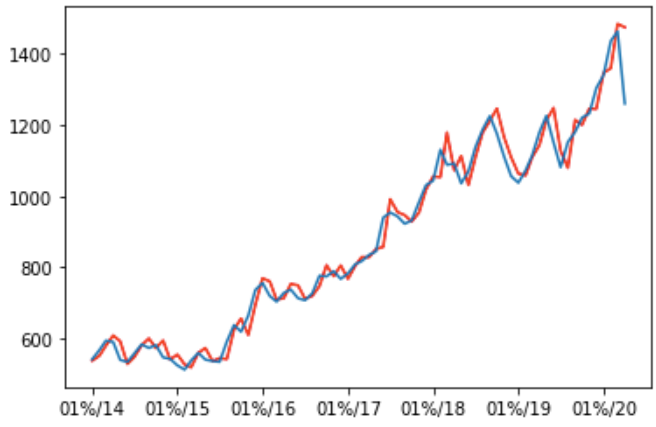
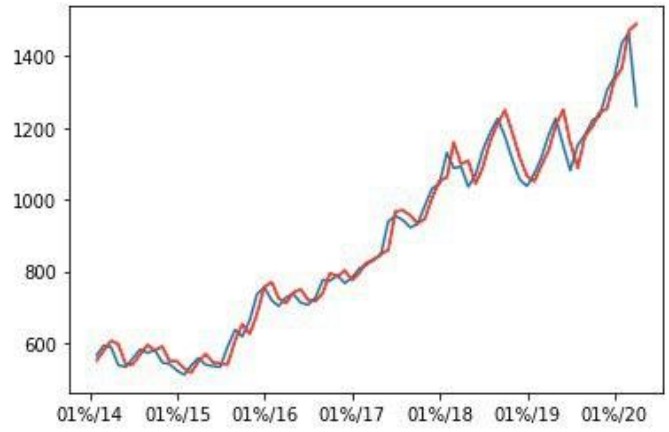


Рисунок 4.11: Google



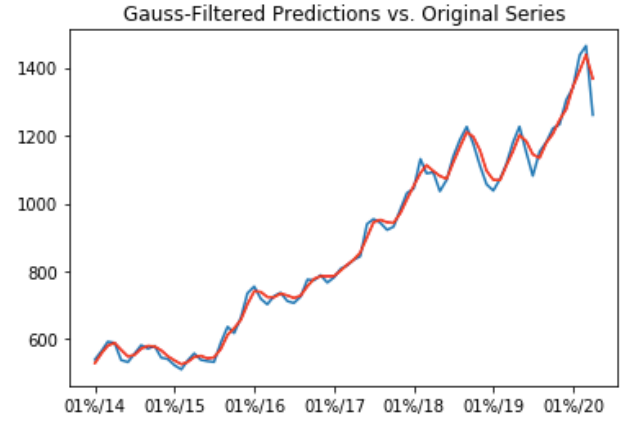
Test RMSE: 44.031

Рисунок 4.12: ARIMA Google



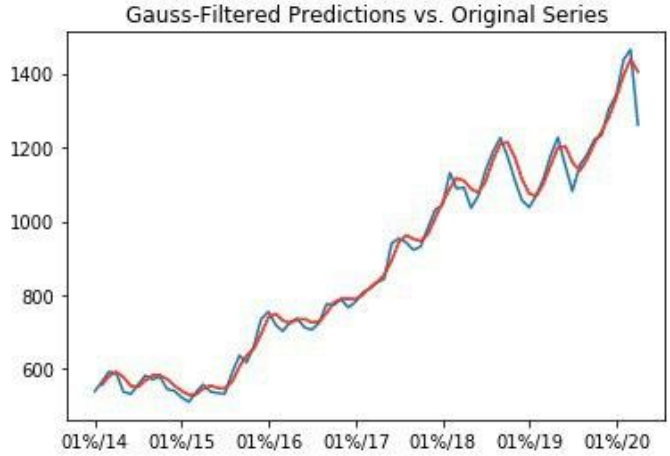
Train RMSE: 18.621
Test RMSE: 45.734

Рисунок 4.13: LSTM Google



Test RMSE: 23.890

Рисунок 4.14: ARIMA Gaussian Filter Google



Test RMSE: 30.488

Рисунок 4.15: LSTM Gaussian Filter Google

Модель ARIMA працює трішки краще, ніж модель LSTM. Гаусова фільтрація даних підвищила точність для обох моделей. Підсумок значень RMSE включає:

ARIMA RMSE (no filter): 44.031

LSTM RMSE (no filter): 45.734

ARIMA RMSE (filtered): 23.890

LSTM RMSE (filtered): 30.488

Після проведення аналізу всі результати були записані у табличку. Як можна побачити з використанням фільтра Гауса RMSE значно покращувалося. Загалом обидва методи дають непогані результати і з не значною різницею. Різкі відмінності простежувалися лише у Eurovision.

Дані	**RMSE ARIMA**	**RMSE LSTM**
Google	44.031	45.734
Google(filtered)	23.890	30.488
Virus Zika	12.727	13.966
Virus Zika (filtered)	6.677	9.717
Eurovision	21.458	49.699
Eurovision (filtered)	12.044	15.088
Game of Thrones	16.478	16.605
Game of Thrones (filtered)	5.874	11.053
Lionel Messi	12.922	15.150
Lionel Messi (filtered)	6.901	8.709
Iphone	11.378	13.729
Iphone (filtered)	6.203	7.603

Висновки

Методи статистичного аналізу є універсальними і можуть застосовуватися в самих різних галузях людської діяльності. Прогнозування часових рядів - це область, яка швидко розвивається, і як така дає багато можливостей для майбутніх робіт.

В цілому результати показують, що і ARIMA, і LSTM є якісними алгоритмами прогнозування даних часових рядів. Модель ARIMA мала меншу кількість помилок, але також може зазнати помилок конвергенції для рядів з різкими градієнтами. LSTM може тренувати модель та робити прогнози для будь-яких даних.

Крім того, Гауссова фільтрація набору даних покращувала прогнози у кожному випадку.

Додатки

```
def arima_model(series, data_split, params, future_periods, log):

    # log transformation of data if user selects log as true
    if log == True:
        series_dates = series.index
        series = pd.Series(np.log(series), index=series.index)

    # create training and testing data sets based on user split fraction
    size = int(len(series) * data_split)
    train, test = series[0:size], series[size:len(series)]
    history = [val for val in train]
    predictions = []

    # creates a rolling forecast by testing one value from the test set, and then add that test value
    # to the model training, followed by testing the next test value in the series
    for t in range(len(test)):
        model = ARIMA(history, order=(params[0], params[1], params[2]))
        model_fit = model.fit(disp=0)
        output = model_fit.forecast()
        yhat = output[0]
        predictions.append(yhat[0])
        obs = test[t]
        history.append(obs)

    # forecasts future periods past the input testing series based on user input
    future_forecast = model_fit.forecast(future_periods)[0]
    future_dates = [test.index[-1]+timedelta(i*365/12) for i in range(1, future_periods+1)]
    test_dates = test.index

    # if the data was originally log transformed, the inverse transformation is performed
    if log == True:
        predictions = np.exp(predictions)
        test = pd.Series(np.exp(test), index=test_dates)
        future_forecast = np.exp(future_forecast)

    # creates pandas series with datetime index for the predictions and forecast values
    forecast = pd.Series(future_forecast, index=future_dates)
    predictions = pd.Series(predictions, index=test_dates)

    # generates plots to compare the predictions for out-of-sample data to the actual test values
    fig = plt.figure()
    ax = fig.add_subplot(111)
    myFmt = mdates.DateFormatter('%m%/%y')
    ax.xaxis.set_major_formatter(myFmt)
    plt.plot(predictions, c='red')
    plt.plot(test)
```

Рисунок 3.1 Реалізація моделі ARIMA

```
def lstm_model(data_series, look_back, split, transforms, lstm_params):
    np.random.seed(1)

    # creating the training and testing datasets
    X_train, y_train, X_test, y_test, train_dates, test_dates, scaler = create_dataset(data_series, look_back, split, transforms)

    # training the model
    model = Sequential()
    model.add(LSTM(lstm_params[0], input_shape=(1, look_back)))
    model.add(Dense(1))
    model.compile(loss='mean_squared_error', optimizer='adam')
    model.fit(X_train, y_train, epochs=lstm_params[1], batch_size=1, verbose=lstm_params[2])

    # making predictions
    train_predict = model.predict(X_train)
    test_predict = model.predict(X_test)

    # inverse transforming results
    train_predict, y_train, test_predict, y_test = \
    inverse_transforms(train_predict, y_train, test_predict, y_test, data_series, train_dates, test_dates, scaler)

    # plot of predictions and actual values
    fig = plt.figure()
    ax = fig.add_subplot(111)
    myFmt = mdates.DateFormatter('%m%/%y')
    ax.xaxis.set_major_formatter(myFmt)
    plt.plot(y_test)
    plt.plot(test_predict, color='red')
    plt.show()

    # calculating RMSE metrics
    error = np.sqrt(mean_squared_error(train_predict, y_train))
    print('Train RMSE: %.3f' % error)
    error = np.sqrt(mean_squared_error(test_predict, y_test))
    print('Test RMSE: %.3f' % error)

    return train_predict, y_train, test_predict, y_test
```

Рисунок 3.2 Реалізація методу LSTM

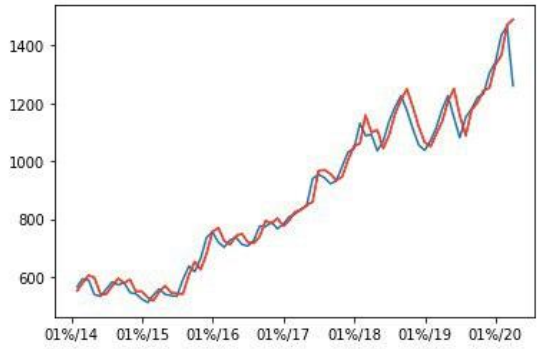
Google LSTM model

```
look_back = 1
split = 0.6
log = True
difference = True
transforms = [log, difference]

nodes = 4
epochs = 5
verbose = 0 # 0=print no output, 1=most, 2=less, 3=least
lstm_params = [nodes, epochs, verbose]

train_predict, y_train, test_predict, y_test = lstm_model(google_ts, look_back, split, transforms, lstm_params)
```

	t-0	t-1
Date		
2019-11-30	0.504599	0.410485
2019-12-31	0.446694	0.504599
2020-01-31	0.529892	0.446694
2020-02-29	0.429111	0.529892
2020-03-31	0.089497	0.429111



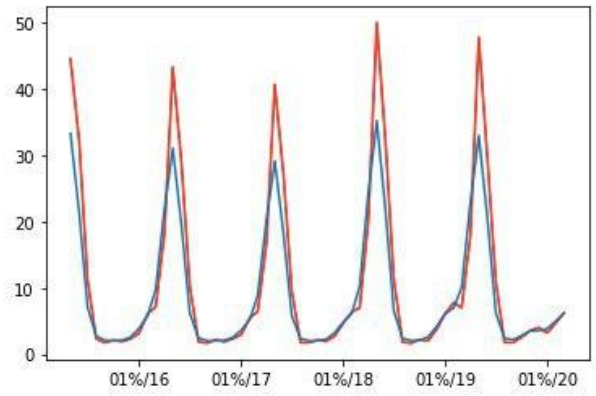
Train RMSE: 18.621
Test RMSE: 45.734

Рисунок 4.1 Приклад роботи програми на даних Google

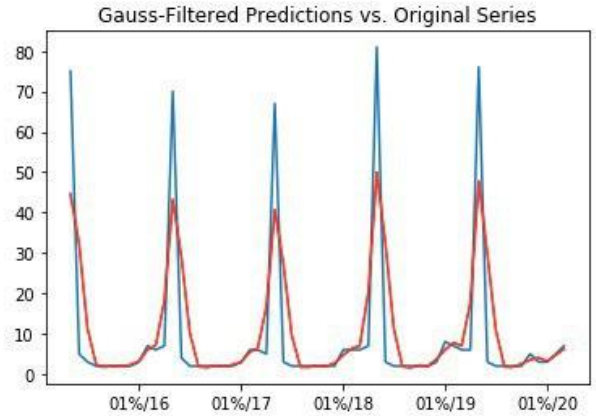
```
# running ARIMA model with Gaussian Filter
data_split = 0.7
p = 1
d = 1
q = 1
params = [p, d, q]
future_periods = 12
log = True

predictions, test, forecast = arima_model(eurovision_gauss, data_split, params, future_periods, log)

# comparing ARIMA model with Gaussian filter to original series
gauss_compare(eurovision, predictions, data_split)
```



Test RMSE: 5.085



Test RMSE: 12.044

Рисунок 4.2 Приклад роботи програми на даних Євробачення

Тема: Прогнозування та аналіз часових рядів

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу.	17.10.2019	
2.	Огляд технічної літератури за темою роботи.	10.11.2019	
3.	Виконати аналіз сучасних методів прогнозування часових рядів	29.11.2019	
3.	Дослідити метод ARIMA та нейронні мережі LSTM	30.12.2019	
4.	Написання першого розділу.	15.01.2020	
5.	Застосування розробленого алгоритму до даних	15.02.2020	
6.	Виконання порівняльного аналізу результатів прогнозування, отриманих за допомогою розробленого алгоритму на основі нейронної мережі LSTM , та результатів, отриманих за допомогою ARIMA моделей.	10.03.2020	
7.	Написання пояснювальної роботи.	15.03.2020	
8.	Створення слайдів для доповіді та написання доповіді.	17.03.2020	
8.	Аналіз отриманих результатів з керівником, написання доповіді та попередній захист курсової роботи.	20.03.2020	
10.	Корегування роботи за результатами попереднього захисту.	28.03.2020	
11.	Остаточне оформлення пояснювальної роботи та слайдів.	12.05.2020	
12.	Захист курсової роботи	24.04.2020	

Студент Женчак А.В

Керівник Щестюк Н.Ю

“_____”_____

Список літератури

- M. Alonso, C. Garcia-Martos, “Time Series Analysis — Forecasting with ARIMA models,” Universidad Carlos III de Madrid, Universidad Politecnica de Madrid. 2012.
- Brockwell, P. and R. Davis (2002). An Introduction to Time Series and Forecasting. Second Edition. Springer-Verlag.
- Chatfield, C. (2004). The Analysis of Time Series: An Introduction. Sixth Edition. Chapman and Hall.
- Edureka. Time Series in Python. https://www.youtube.com/watch?v=e8Yw4alG16Q&t=181s
- J. Brownlee, “How to Create an ARIMA Model for Time Series Forecasting with Python,” https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/, 2017
- James Chen “Autoregressive Integrated Moving Average (ARIMA)” https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp
- Robert H. Shumway, David S. Stoffer (2011) Time Series Analysis and Its Applications With R Examples
- Sharmistha Chatteerjee “ARIMA/SARIMA vs LSTM with Ensemble learning Insights for Time Series Data“ https://towardsdatascience.com/arima-sarima-vs-lstm-with-ensemble-learning-insights-for-time-series-data-509a5d87f20a
- Selva Prabhakaran“ARIMA Model – Complete Guide to Time Series Forecasting in Python” https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/
- А. Ю. Лоскутов. Анализ временных рядов. Курс лекций https://chaos.phys.msu.ru/loskutov/PDF/Lectures_time_series_analysis.pdf
- Анализ временных рядов и прогнозирование: Учебник. — М.: Финансы и статистика, 2001. — 228 с.: ил.
- Афанасьев В.Н., Юзбашев М.М. А94 Анализ временных рядов и прогнозирование: Учебник. — М.: Финансы и статистика, 2001. — 228 с.: ил.

- Бокс Дж. Анализ временных рядов. Прогноз и управление: Вып. 1 / Дж. Бокс, Г. Дженкинс. – М.: Мир, 1974. – 406 с.
- Бююль А. SPSS: искусство обработки информации. Анализ статистических данных и восстановление скрытых закономерностей/ А.Бююль, П.Цефель; Под ред. В.Е.Момота.- СПб.: ДиаСофтИП, 2002.- 608с
- Вільям Г. Грін. Економетричний аналіз / Пер. з англ. А. Олійник, Р. Ткачук.; Наук. ред. пер. О. Комашко; Передм. О. І. Черняка, О. В. Комашко. — К.: Видавництво Соломії Павличко «ОСНОВИ», 2005. — 1197 с
- Временной ряд- модель LSTM https://coderlessons.com/tutorials/mashinnoe-obuchenie/izuchit-vremennye-riady/vremennoi-riad-model-lstm
- Інформаційна технологія прийняття рішень на основі прогнозування часових рядів з подвійною довгою пам’яттю: монографія / за заг. ред. Р. Н. Квєтного. – Вінниця : ВНТУ, 2012. – 140 с.
- Канторович Г.Г. Анализ временных рядов / Г.Г. Канторович // Экономический журнал ВШЭ. 2002. – Т. 6